ISSN 0868-6157

# КОМПЬЮТЕР ПРЕСС





11'92

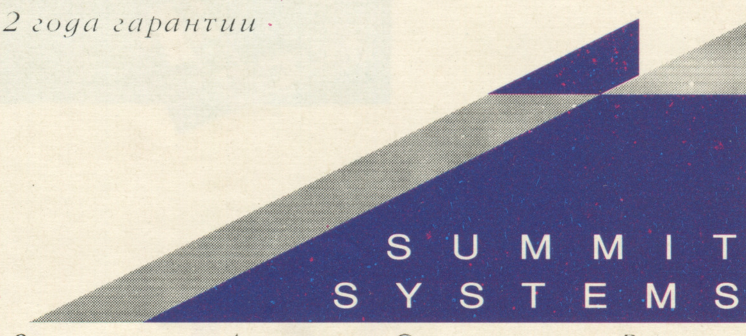
# Поставьте будущее себе на стол!




- Ваши потребности растут? Наш компьютер совершенствуется!
- Надежность под знаком Intel, CHIPS, Quantum, Sony
- Уникальная производительность - убедитесь сами
- 2 года гарантии обеспечены быстрым обслуживанием в 50-ти городах от Бреста до Находки
- Комплексный подход: 286-е компьютеры, принтеры, сетевое оборудование и программное обеспечение

## Модульный Саммит

*386-33/ 486sx-25/ 486-40MHz*
*цветной Super VGA монитор (1024x768)*
*ОЗУ 4 Мб (макс. 32 Мб)*
*Жесткий диск 105/210 Мб (17/15 мс)*
*Русифицированная клавиатура*
*Лицензионная ДОС*
*2 года гарантии*




*Задумано в Америке. Сделано для России.*

МОСКВА (095) 299 1162
ВНЕ МОСКВЫ (0172) 973 119

# КОМПЬЮТЕР ПРЕСС




11'92




КОМПЬЮТЕРПРЕСС

Издается с 1989 года
Выходит 12 раз в год
11'92 (35)



**Адрес редакции:**
*113093 Москва, аб.ящик 37*
*Факс: (095) 200-22-89*
*Телефон для справок: (095) 471-32-63*
*E-mail: editorial@computerpress.msk.su*

Сдано в набор 5.09.92. Подписано к печати 15.10.92.
Формат 84х108/16. Печать офсетная. Бумага типографская. Усл.печ.листов 8,4+0,42 (обл.).
Тираж 50000 экз. Заказ 3138. С-11.

Оригинал-макет подготовлен агентством «КомпьютерПресс».

Тексты проверены системой «ОРФО».

Журнал сверстан на оборудовании фирмы Summit Systems.

Отпечатано в полиграфической фирме «Красный пролетарий» РГИИЦ «Республика».
103473 Москва, И-473, Краснопролетарская, 16.




*Мы вновь обращаемся к нашей рубрике для наиболее любознательных читателей, в которой более или менее подробно рассказываем о принципах функционирования различных узлов компьютера и его периферийных устройств. Сегодня речь пойдет о сканерах.*

# Сканеры без пролога и эпилога

"Если это и не правда, то хорошо придумано"
**древнеримский афоризм**

## Вместо пролога

Используете ли вы сканер в своей работе или просто знаете о его существовании, вы наверняка не раз задавались вопросом о том, каков же принцип работы этого устройства. Часто, к сожалению, вместо ответа на него приходится довольствоваться полузабытым детским софизмом: "Чем меньше знаешь, тем меньше забываешь, а чем меньше забываешь, тем больше знаешь", поскольку имеющийся в распоряжении User´s Guide (в лучшем случае) вопросы функционирования устройств, как известно, не освещает. Нельзя сказать, что информация о сканерах в доступной технической литературе полностью отсутствует, однако принцип работы того или иного сканера, представляющий закономерный интерес, часто скромно умалчивается. Попробуем в какой-то мере восполнить это упущение.

Хотя автор этих строк примерно представляет, как работает сканер, по крайней мере черно-белый (где-то когда-то что-то читал), однако он не может набраться нахальства и заявить, что статья о принципах работы сканеров (особенно цветных) — плод его "ума холодных наблюдений и сердца горестных замет". Для уяснения принципов работы (особенно цветных) сканеров ему пришлось обратиться к "гуманитарной помощи" — журналам BYTE и PC Magazine. Великий Кормчий, правда, утверждал, что если "много книг прочтешь — Императором не станешь", но мы то с вами, уважаемый читатель, в наши беспечально-демократические времена к этому и не стремимся. Для начала ограничимся определением того, что же такое сканер.

Сканером называется устройство, позволяющее вводить в компьютер образы изображений, представленных в виде текста, рисунков, фотографий или другой графической информации. Это определение, разумеется, не блещет изысканностью форм, но, пожалуй, не хуже определения телевизора, которое услышали выпускники радиоинститута на военной переподготовке. "Телевизор — это набор радиодеталей, соединенных определенным образом так, что на экране появляется изображение". Но шутки в сторону!

## Какие они бывают?

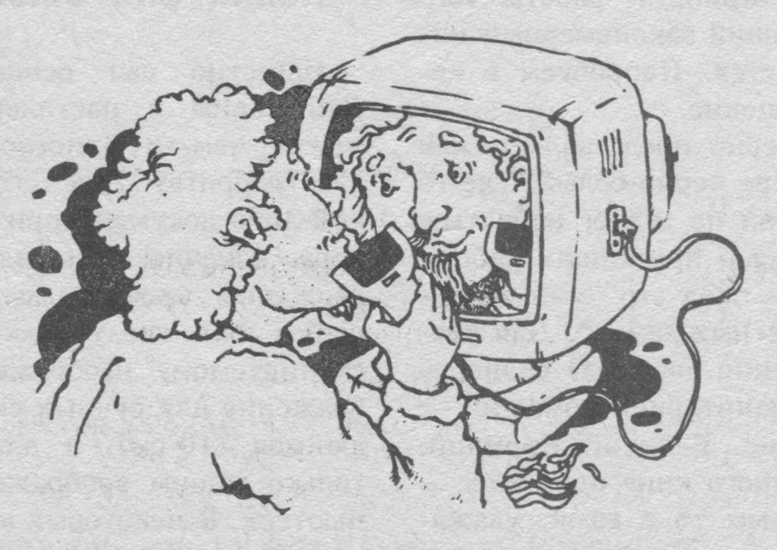
Известно два основных типа сканеров: ручной (hand-held) и настольный (desktop). Ручной сканер может чем-то напоминать увеличенную в размерах электробритву. Для того чтобы ввести в компьютер какой-либо документ при помощи этого устройства (сканера, конечно, а не электробритвы), надо без резких движений провести сканирующей головкой (находящейся там, где у электробритвы обычно "ножи") по сканируемому изображению. Ширина вводимого изображения для ручных сканеров не превышает обычно 4 дюймов (10 см), а длина, очевидно, ограничивается только вашим воображением и объемом памяти компьютера. В некоторых моделях ручных сканеров, в уго-

ду повышения разрешающей способности, уменьшают ширину вводимого изображения. Надо сказать, что ручные сканеры — вещь, в некоторых случаях достаточно удобная. Например, читатели немецкого журнала "Magazin fur ComputerPraxis" имеют возможность вводить в свой компьютер информацию прямо с "бумажного диска" — специальных страниц этого журнала, на которых подобная информация зашифрована определенным образом. И все-таки, в общем случае, касаясь эффективности ручных сканеров, автор этих строк от комментариев бы воздержался.

Другое дело настольные сканеры. Их у нас, кстати, называют и страничными, и планшетными, и даже автосканерами(?). Такие сканеры позволяют обычно вводить изображения размерами 8,5 на 11 или 8,5 на 14 дюймов. Существует три разновидности настольных сканеров: flatbed, sheet-fed и overhead. Надо сказать, что на нашем скудном компьютерном рынке настольные сканеры представлены несколько однобоко. Не ошибусь, если скажу, что подавляющее число таких сканеров — это различные модели ScanJet глубокоуважаемой всеми фирмы Hewlett-Packard. Все эти сканеры относятся, как правило, к наиболее распространенной разновидности — flatbed-сканерам.

Flatbed-сканеры — обычно достаточно дорогие устройства, но, пожалуй, и наиболее "способные". Внешне они чем-то могут напоминать копировальные машины — "ксероксы", внешний вид которых известен, конечно, многим. Для сканирования изображения (чего-нибудь) необходимо открыть крышку сканера, положить сканируемый лист на стеклянную пластину изображением вниз, после чего закрыть крышку. Все дальнейшее управление процессом сканирования осуществляется с клавиатуры компьютера — при работе с одной из специальных программ, поставляемых вместе с таким сканером. Понятно, что рассмотренная конструкция сканера позволяет (подобно "ксероксу") сканировать не только отдельные листы, но и страницы журнала или книги.

Работа sheet-fed-сканеров чем-то напоминает работу обыкновенной факс-машины. Отдельные листы документов протягиваются через такое устройство, при этом и осуществляется их сканирование. Понятно, конечно, что в этом случае копирование страниц книг и журналов просто невозможно. Рассматриваемые сканеры достаточно широко используются в областях, связанных с оптическим распознаванием символов (Optical Character Recognition, OCR). Для удобства работы sheet-fed-сканеры обычно оснащаются устройствами для автоматической подачи страниц.



Третья разновидность настольных сканеров — overhead-сканеры, которые больше всего напоминают несколько своеобразный overhead-проектор. Вводимый документ кладется на поверхность сканирования изображением вверх, блок сканирования находится при этом также сверху.

Вот, пожалуй, и все, что можно сказать о типах и конструктивных особенностях известных на сегодня сканеров.

## "Часто простое кажется сложным..."

В настоящее время, когда говорят о сканере, то имеют в виду обычно либо черно-белый, либо цветной сканер, причем под черно-белым сканером понимают сканер, воспринимающий оттенки серого цвета, — так называемый полутоновой (gray-scale) сканер. Но было это, конечно, не всегда.

Первые модели черно-белых сканеров могли работать только в двухуровневом (bilevel) режиме, воспринимая или черный, или белый цвет. Таким образом, сканироваться могли только либо штриховые рисунки (например, чертежи), либо двухтоновые изображения. Хотя эти сканеры и не могли работать с действительными оттенками серого цвета, выход для сканирования полутоновых изображений такими сканерами был найден. Псевдополутоновой режим (dithering) работы такого сканера имитирует оттенки серого цвета, группируя несколько точек вводимого изображения в так называемые gray-scale-пикселы. Такие пикселы могут иметь размеры 2x2 (4 точки), 3x3 (9 точек) или 4x4 (16 точек) и т.д. Отношение количества черных точек к белым и определяет уровень серого цвета. Например, gray-scale-пиксел размером 4x4 позволяет воспроизводить 17 уровней серого (включая и полностью белый цвет). Не следует, правда, забывать, что разрешающая способность сканера при использовании gray-scale-пикселов снижается (в последнем случае — в 4 раза).

Полутоновые сканеры используют максимальную разрешающую способность, как правило, только в двухуровневом режиме. Обычно такие сканеры поддерживают 16, 64 или 256 оттенков серого цвета, для 4-, 6- и 8-разрядного кода, который ставится при этом в соответствие каждой точке изображения. Разрешающая способность сканера измеряется в количестве различаемых точек на дюйм изображения — dpi (dot per inch). Если в первых моделях сканеров разрешающая способность была обычно 200-300 dpi, то в современных моделях это, как правило, 400, а то и 800 dpi. Обычно разрешение сканера может устанавливаться программным путем в процессе работы из ряда значений: 75, 100, 150, 200, 300 и 400 dpi.

Надо сказать, что благодаря операции интерполяции, выполняемой обычно про-




**ПРЕДСТАВЛЯЕТ**
оригинальные программные продукты для IBM-совместимых компьютеров:

**А/О "Мир - Диалог"**
**Россия, 115598, Москва, ул. Ягодная, 17, комн. 69**
**Тел.: (7-095) 329-46-88,**
**Факс: (7-095) 329-47-11**

## FFT-PRO (L-master)

Преобразование аналоговых сигналов в цифровые с частотой 70 KHz (250 KHz) с многоканальным анализом и графическим отображением характеристик сигнала. Используется в медицине, механике, геологии, геофизике, при проведении гидроакустических исследований и для автоматического контроля и управления системами непрерывного и дискретного действия.

## ЗВУК-12

Средство, обеспечивающее высококачественное звуковое сопровождение Ваших прикладных программ. Ввод с микрофона, редактирование, упаковка и воспроизведение звуковых фрагментов.

## ArtVox

Двухмануальный многоголосый музыкальный синтезатор. Ритмический компьютер, позволяющий использовать до 9 ударных и мелодических инструментов. Музыкальный проигрыватель с 8-канальным микшерским пультом.

## PYGMALION

Создание и воспроизведение озвученных мультфильмов. Оригинальный алгоритм позволяет быстро воспроизводить динамику изображения. Мультфильм озвучивается с помощью встроенного музыкального редактора.

## SpaceWriter

Создание броских рекламных плакатов с моделированием пространственного размещения изображений и текстов.

## VectorVision

Высокоскоростное преобразование графического поточечного изображения в векторное представление. Работает в среде Windows. На входе картинки в формате: TIFF 5.0, GIF, PCX, RLC, BMP, RLE, DIB, TARGA. На выходе один из форматов: DXB, DXF, EPS, WMF, SLD, XS.

## XorGraphics

Библиотека процедур компьютерной графики для профессионалов. Обеспечивает эффективную реализацию графики по скорости и по размерам объектных модулей.

## OneStep

Инструмент для создания графического пользовательского интерфейса. Реализован в виде библиотеки процедур для языка Си.

## OTLFont

Инструмент, позволяющий создавать векторные шрифты для WINDOWS, HALO, WORKS, TURBO.

## АРТ

Автоматизированное Распознавание Текста со сканера. Распознаются печатные тексты на русском, английском и испанском языках. "АРТ - лучший пакет для тех, кто не собирается проблему распознавания сделать делом своей жизни, а хочет просто вводить документы в компьютер".

## ДИАКОР

Проверка орфографии русского языка. Встроенный словарь содержит 3 миллиона словоформ. Прост и надежен в обращении.

## TRANSFER

Резидентный русско-английский и англо-русский словарь. Есть модификация для французского языка.

## CYRILLIC

Поддержка русского алфавита для клавиатуры и адаптеров EGA, VGA.

## PC TELETEXT

Оперативная деловая информация со всего мира, передаваемая по системе TELETEXT, на Вашем компьютере. Информация включает в себя цены на товары и услуги, курсы валют, обзоры и прогнозы, политические новости. PC Teletext декодирует входной видеосигнал и вводит текстовую информацию в IBM-совместимый компьютер. Входным сигналом служит видеосигнал обычной телетрансляции либо с тюнера спутникового телевидения (Super Channel, TRT Int., Rai, TVE Int., Discovery Channel).

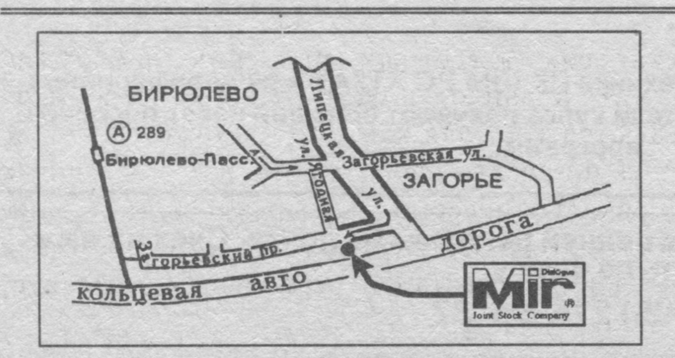
## ACE IMMUNE SYSTEM

Антивирус. Избавляет от проблем борьбы с компьютерными вирусами. Антивирус прививается программам, которые нужно защитить от вируса. Защищает от "бутовых" вирусов и вылечивает зараженные файлы.

## ВЕБЕР

Медицинская справочно-консультативная система для контроля за здоровьем женщин: диеты, косметика, лекарственные растения, домашняя аптека. Все о беременности, включая прогноз состояния и ведение текущей и отчетной документации. Используется в центрах здоровья, женских консультациях, медицинских учебных заведениях, поликлиниках.

**Программные средства западных фирм ( Microsoft, Borland)**




**Проезд:**
м.Царицыно, от центра выход из головного вагона направо, авт. 203, 701 или
ст.Бирюлево-Пасс., авт. 289, 690.
Остановка "Загорьевская, 10"

**Реквизиты:**
А/О "Мир-Диалог", п/и банка 103012, Ком.банк "ДиалогБанк" г.Москва, корр.сч. 161625 в РКЦ ГУ ЦБР, N 467073, МФО 201791, уч-к 83

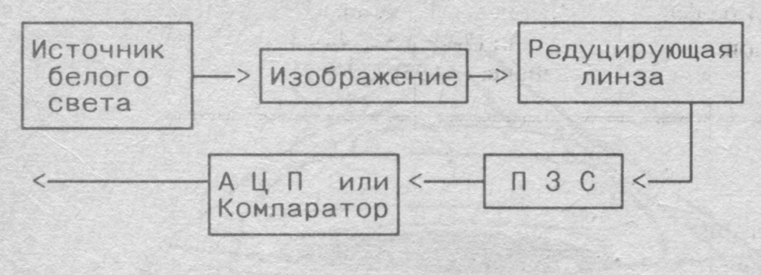



Рис. 1

граммно, современные сканеры могут иметь разрешение 800 и даже 1600 dpi. В результате интерполяции на получаемом при сканировании изображении сглаживаются кривые линии и исчезают неровности диагональных линий. Для тех кто не помнит, поясним. Интерполяция позволяет отыскивать значения промежуточных величин по уже известным значениям. Например, в результате сканирования один из пикселов имеет значение уровня серого цвета 48, а соседний с ним — 76. Использование простейшей линейной интерполяции позволяет сделать предположение о том, что значение уровня серого цвета для промежуточного пиксела могло бы быть равно 62 [(48+76)/2]. Если вставить все оценочные значения пикселов в файл отсканированного изображения, то разрешающая способность сканера как бы удвоится, то есть вместо обычных 400 dpi станет равной 800 dpi. Тут бы и выразить благодарность господам Ньютону и Лагранжу!

## "...черное — белым, белое — черным"

На рис. 1 приведена примерная блок-схема, отражающая принцип работы черно-белого сканера. Сканируемое изображение освещается белым светом, получаемым либо от флюоресцентной лампы, либо от лампы накаливания, раскаляемой добела. Отраженный свет через редуцирующую (уменьшающую) линзу попадает на фоточувствительный полупроводниковый элемент, называемый Прибором с Зарядовой Связью — ПЗС (Charge-Coupled Device, CCD). Каждая строка сканирования изображения соответствует определенным значениям напряжения на ПЗС. Эти значения напряжения преобразуются в цифровую форму либо через аналого-цифровой преобразователь АЦП (для полутоновых сканеров), либо через компаратор (для двухуровневых сканеров). Компаратор сравнивает два напряжения (от ПЗС и опорное), причем в зависимости от результата сравнения на его выходе формируется сигнал 0 (черный цвет) или 1 (белый). Разрядность АЦП для полутоновых сканеров зависит от количества поддерживаемых уровней серого цвета. Например, сканер, поддерживающий 64 уровня серого, должен иметь 6-разрядный АЦП.

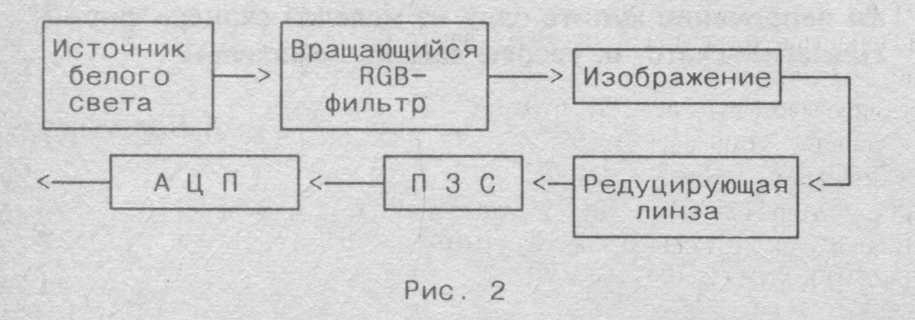



Рис. 2

Каким образом сканируется каждая следующая строка изображения, целиком зависит от типа используемого сканера. У flatbed-сканеров движется сама сканирующая головка. В sheet-fed-сканерах головка остается неподвижной, а движется носитель с изображением — бумага. Overhead-сканеры используют движение отражающего зеркала.

## Один источник на три составные части

В настоящее время существует несколько технологий для получения цветных сканируемых изображений. На рис. 2 представлена блок-схема, поясняющая один из наиболее общих принципов работы цветного сканера. Сканируемое изображение освещается уже не белым светом, а через вращающийся RGB-светофильтр. Для каждого из основных цветов (красного, зеленого и синего) последовательность операций практически не отличается от последовательности операций при сканировании черно-белого изображения. Исключение составляет, пожалуй, только этап предварительной обработки и гамма-коррекции цветов, перед тем как информация передается в компьютер. Понятно, что этот этап является общим для всех цветных сканеров.

В результате трех проходов сканирования получается файл, содержащий образ изображения в трех основных цветах — RGB (образ композитного сигнала). Если используется 8-разрядный АЦП, который поддерживает 256 оттенков для одного цвета, то каждой точке изображения ставится в соответствие один из 16,7 миллиона возможных цветов (24 разряда). Сканеры, использующие подобный принцип действия, выпускаются, например, фирмой Microtek.

Надо отметить, что наиболее существенным недостатком описанного выше метода является увеличение времени сканирования в три раза. Проблему может представлять также “выравнивание“ пикселов при каждом из трех проходов, так как в противном случае возможно размывание оттенков и “смазывание“ цветов.

## Цвет, который идет с Востока

В сканерах известных японских фирм Epson и Sharp, как правило, вместо одного источника света используются три, для каждого цвета отдельно. Это позволяет сканировать изображение всего за один проход и исключает неверное “выравнивание“ пикселов. Сложности этого метода заключаются обычно в подборе источников света со стабильными характеристиками.

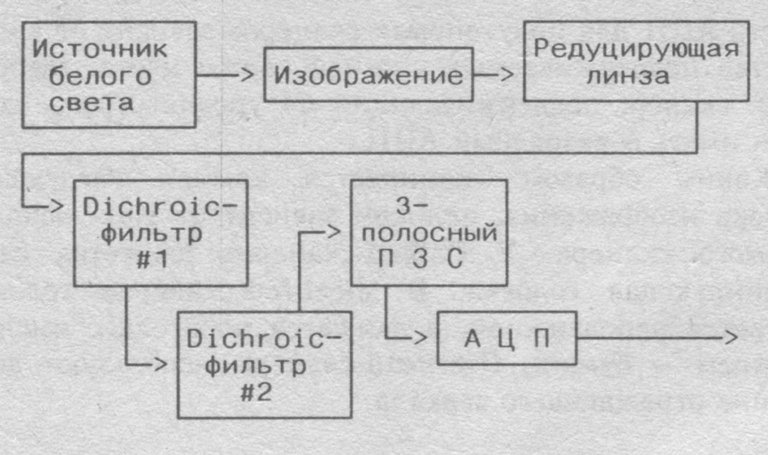



Рис. 3




Другая японская фирма Seiko Instruments разработала цветной flatbed-сканер SpectraPoint, в котором элементы ПЗС были заменены фототранзисторами. На ширине 8,5 дюймов размещено 10 200 фототранзисторов, которые расположены в три колонки по 3 400 в каждой. Три цветных фильтра (RGB) расположены так, что каждая колонка фототранзисторов воспринимает только один основной цвет. Высокая плотность интегральных фототранзисторов позволяет достигать хорошей разрешающей способности — 400 dpi (3 400/8,5) — без использования редуцирующей линзы. Применение такой технологии делает сканер SpectraPoint, пожалуй, самым быстрым из всех существующих.

## Находка от Hewlett-Packard

На рис. 3 представлена блок-схема, поясняющая принцип действия цветного сканера ScanJet IIc фирмы Hewlett-Packard. Источник белого света освещает сканируемое изображение, а отраженный свет через редуцирующую линзу попадает на трехполосную ПЗС через систему специальных dichroic-фильтров, которые и разделяют белый свет на три компоненты: красную, зеленую и синюю. Физика работы dichroic-фильтров связана с явлением дихроизма, заключающегося в различной окраске одноосных кристаллов в проходящем белом свете, в зависимости от положения оптической оси. В рассматриваемом случае фильтрация осуществляется парой dichroic-фильтров, каждый из которых представляет собой "сэндвич" из двух тонких и одного более толстого слоя кристаллов. Первый слой первого фильтра отражает синий свет, но пропускает зеленый и красный. Второй слой отражает зеленый свет и пропускает красный, который отражается только от третьего слоя. Во втором фильтре, наоборот, от первого слоя отражается красный свет, от второго — зеленый, а от третьего — синий. После системы фильтров разделенные красный, зеленый и синий свет попадают на собственную полосу ПЗС, каждый элемент которого имеет размер около 8 мкм. Дальнейшая обработка сигналов цветности практически не отличается от обычной. Заметим, что подобный принцип работы (с некоторыми отличиями, разумеется) используется и в цветных сканерах фирмы Ricoh.

## Вместо эпилога

Как правило, образы изображений в компьютере хранятся в графических файлах формата TIFF (Tagged Image File Format) или в файлах не менее популярного формата PCX. Надо иметь в виду, что при сканировании изображений файлы получаются достаточно громоздкие. Так, при вводе полутонового черно-белого изображения размером 8 на 10 дюймов с 256 оттенками серого цвета и при разрешающей способности 400 dpi будет создан файл размером более 12 Мбайт. Для уменьшения объема хранимой информации используется обычно сжатие таких графических файлов.

В заключение скажем несколько слов и об интерфейсах сканеров. Ручные сканеры, как правило, используют специальную 8-разрядную интерфейсную плату, вставляемую в слот расширения. Часто такие платы требуют свободных каналов прерывания IRQ и/или прямого доступа в память DMA. В настольных сканерах наиболее распространенным интерфейсом является SCSI, хотя в некоторых случаях сканер может подключаться и через последовательный порт RS-232.

Как уже отмечалось выше, проблемы выбора при покупке сканера у нас в стране пока не существует — вы непременно купите одну из моделей сканера фирмы Hewlett-Packard, и, уверяю вас, не пожалеете!

*А.Борзенко*



# Следовательно — Sequent!

В конце 1991 года, после отмены ограничений КОКОМ, на наш рынок вышла американская фирма Sequent Computer Systems — ведущий производитель симметричных многопроцессорных систем с операционной системой UNIX.

Эта молодая (основана в 1983 г.), динамично развивающаяся компания с 1986 по 1991 год имела стабильный рост доходов, объема продаж и числа представительств в разных странах мира. В 1991 году доходы фирмы превысили 300 миллионов долларов США. Сегодня она имеет отделения во всех развитых странах мира, насчитывает более 4000 установленных систем своего производства, причем в основном эти машины установлены в таких "избалованных" компьютерами странах, как США, Япония, ФРГ и Великобритания.

Секрет успеха фирмы Sequent кроется в нестандартном сочетании стандартных решений: процессоров i80486, шин SCSI и VMEbus, операционной системы UNIX. При этом специалистам фирмы удалось решить две проблемы: сделать все процессоры полностью симметричными и, что не удавалось еще никому из производителей (насколько известно автору), опровергнуть печально известное правило 80%.

Согласно этому правилу, при добавлении к системе следующего процессора производительность каждого из процессоров составляет не более 80% от производительности на предыдущем этапе (остальные 20% уходят на обеспечение межпроцессорного взаимодействия), из-за этого практически не имеет смысла устанавливать в системе больше 4-5 процессоров. Победа над этим правилом позволяет достичь в системах Sequent рекордного для универсальных ЭВМ числа в 30 процессоров, причем при добавлении новых процессоров производительность систем растет линейно.

Симметричность обеспечивает полную независимость и взаимозаменяемость процессоров и повышенную надежность машин фирмы.

Благодаря этому фирма сумела создать серию мощных (класса от супермини- до суперЭВМ) и относительно дешевых машин, далеко опережающих конкурирующие модели по соотношению стоимость/производительность.

## Архитектура систем Sequent

Все три системы новейшего семейства Symmetry 2000 фирмы Sequent имеют практически одну и ту же архитектуру с разделяемой памятью и общей высокоскоростной системной шиной. На системную шину устанавливаются контроллеры памяти (каждый из которых может иметь плату расширения) и процессорные платы (рис. 1).

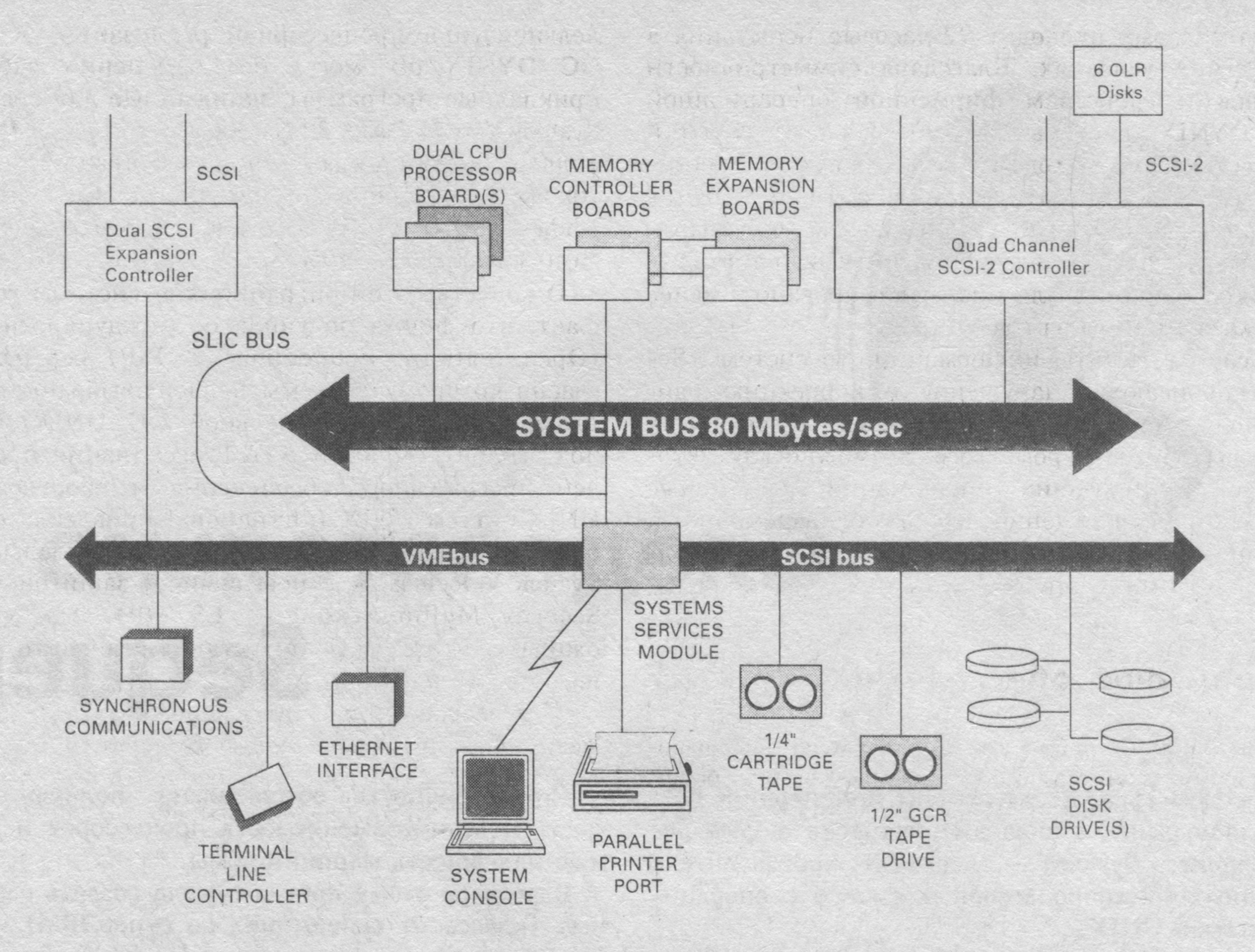
Максимальный объем оперативной памяти и максимальное число процессоров, а также количество поддерживаемой дисковой памяти зависят от модели системы:

Symmetry 2000/2x0: 320 Мбайт, 81 Гбайт, 6 CPU
Symmetry 2000/4x0: 512 Мбайт, 131 Гбайт, 10 CPU
Symmetry 2000/7x0: 960 Мбайт, 260 Гбайт, 30 CPU

На каждой процессорной плате размещено два 32-разрядных процессора i80486, каждый с кэш-памятью на 512 Кбайт. Все процессоры полностью независимы и симметричны, то есть каждый из них может работать как с операционной системой, так и с программным приложением. Кроме того, на процессорных платах установлены так называемые контроллеры системных связей и прерываний, которые



2. Зак. № 3138




взаимодействуют через специальную шину и обеспечивают согласованность работы всех процессоров. В системах серии x00 используются процессоры с частотой 25 МГц, а в системах серии x50 — процессоры с частотой 50 МГц.

К системной шине подсоединены стандартная шина SCSI, к которой подключаются магнитные и оптические диски, стримеры и другие устройства памяти, и шина VMEbus (для моделей 450 и 750 возможно наличие более одной шины), к которой в свою очередь подключаются терминалы, сетевые адаптеры, дополнительные шины SCSI и, возможно, нестандартные устройства. Наконец, через специальную плату к системной шине можно подключать дополнительные шины SCSI-2, которые, как известно, обеспечивают очень быстрый обмен с дисковыми накопителями.

Благодаря своей архитектуре системы Sequent обладают рядом весьма привлекательных для пользователя характеристик, некоторые из них просто уникальны.

## Производительность

Применение в последних моделях Sequent 486-х процессоров с тактовой частотой 50 МГц позволяет при операциях с целыми числами достичь производительности в 41 MIPS на каждый процессор. Например, при выполнении транзакций в режиме реального времени (OLTP) для "тяжеловесной" базы данных с объемом 50 Гбайт под управлением СУБД ORACLE на системе Sequent Symmetry 2000/750 превзойден порог 200 tps (транзакций в секунду), а для более "легкой" СУБД Sybase производительность в 170 tps достигнута на младшей модели Symmetry 2000/250 с четырьмя процессорами.

## Наращиваемость

Наращивание числа процессоров и физической памяти систем Sequent осуществляется простым добавлением соответствующих плат. Модификация программного обеспечения при этом не требуется. При запуске система производит самотестирование и автоматически реконфигурируется. Кроме того, все модели Sequent совместимы на уровне двоичных кодов. Если пользователю недостаточно возможностей роста, заложенных в одной модели, то, поскольку архитектура всех систем практически одинакова, он может с минимальными затратами перейти на следующую модель, взяв из старого корпуса всю "начинку": процессорные платы, платы памяти, диски и т.д. Более того, в одной системе могут быть установлены процессоры и на 25 и на 50 МГц.

## Надежность

Надежность систем Sequent обусловлена высоким качеством исходных процессоров и готовых машин,

каждая из которых проходит 72-часовые испытания в очень жестких условиях. Благодаря симметричности процессоров и свойствам фирменной операционной системы DYNIX, системы Sequent обладают скрытой избыточностью: при выходе из строя одного из процессоров нагрузка автоматически перераспределяется между остальными. При испытаниях в экстремальных условиях механических, электромагнитных и температурных колебаний было зарегистрировано менее одного отказа системы за год.

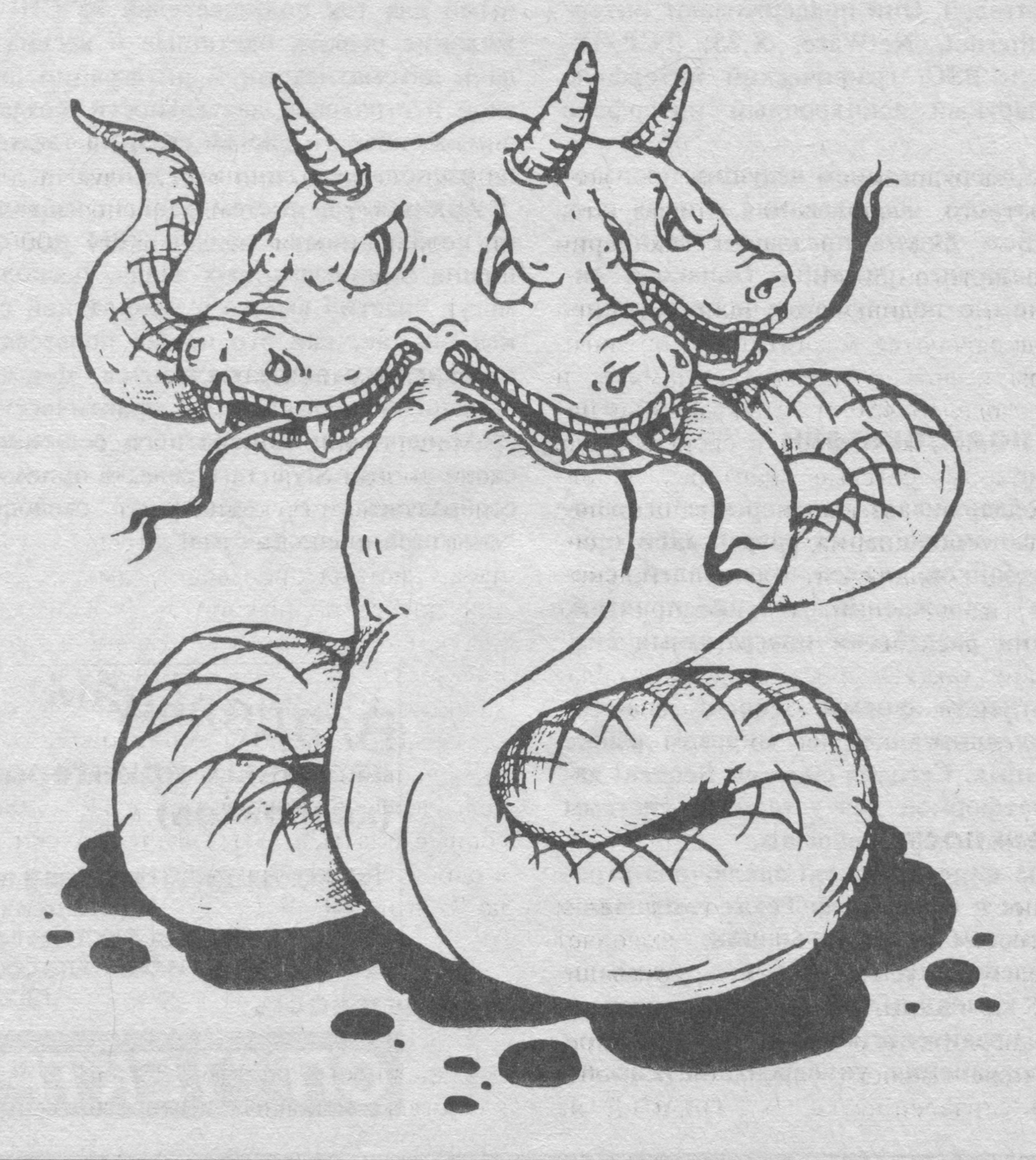
Что касается защиты информации, то системы Sequent обеспечивают обнаружение и коррекцию одиночных неправильных битов и диагностику двойных неправильных битов. Кроме того, возможно как автоматическое дублирование информации на разные диски, так и подключение дисков одновременно к двум машинам, что обеспечивает высокий уровень защиты информации.

## Программное обеспечение

Системы Sequent работают под управлением высокоэффективной и уникальной по своим возможностям операционной системы (ОС) DYNIX/ptx — параллельной многопроцессорной реализации ОС UNIX. В ОС DYNIX/ptx могут без изменений работать все прикладные программы, написанные для систем UNIX System V и UNIX 4.2.bsd. Кроме того, ОС DYNIX/ptx поддерживает стандартный для открытых систем протокол Network File System (NFS), что обеспечивает эффективное взаимодействие систем Sequent с другими UNIX-системами.

О качестве этой операционной системы говорит тот факт, что фирма получила от международного фонда (Open Software Foundation — OSF) сертификат, согласно которому системы Sequent выбраны в качестве платформы для параллельной ОС UNIX. Созданная под эгидой компании AT&T ассоциация производителей программного обеспечения и оборудования для UNIX-систем UNIX International привлекла фирму Sequent к разработке стандарта на параллельный UNIX System V Release 4 с повышенной защитой (Enhanced Security/Multiprocessing — ES/MP). Так что можно ожидать, что DYNIX/ptx станет стандартом де-факто на параллельный UNIX.

Для разработчиков прикладных систем, которые не могут позволить себе купить Sequent, фирма разработала MicroDYNIX/ptx — версию DYNIX/ptx для персональных ЭВМ. Разработанные для нее программы



совместимы с ОС DYNIX/ptx на уровне двоичных кодов.

Из средств программирования ОС DYNIX/ptx поддерживает языки C, C++, Фортран, Кобол, Ада, Паскаль, Лисп, Пролог и Бейсик, а также стандартный язык запросов баз данных SQL.

Архитектура и программное обеспечение систем Sequent гарантируют чрезвычайно эффективное выполнение запросов реляционных баз данных (ORACLE, INGRES, INFORMIX, PROGRESS, UNIFY и др.).

Для ОС DYNIX/ptx разработаны прикладные системы для автоматизации учреждений, банковские системы, системы финансового учета, управления кадрами, информационные и графические системы, автоматизированные системы разработки программного обеспечения, АСУ предприятий и др.

## Интерфейсы и средства связи

Работающие под UNIX системы вообще отличаются открытостью, но даже на этом фоне системы Sequent выделяются разнообразием поддерживаемых средств коммуникации и сопряжения с системами разных архитектур и производителей. Они поддерживают интерфейсы с сетями Ethernet, NetWare, X.25, TCP/IP, NFS, DECnet, SNA и BSC, графический интерфейс X-Windows и стандартный асинхронный интерфейс RS-232.

Для сопряжения с оборудованием ведущих производителей ЭВМ и сетевого оборудования, таких как DEC, IBM, Novell, Sun, фирма предлагает стандартные программно-аппаратные решения. Однако к системам Sequent возможно подключение и нестандартных устройств.

## Партнеры и пользователи

Фирма Sequent поддерживает партнерские отношения с компаниями, занимающимися разработкой прикладных систем для банков, отелей, госпиталей, систем автоматизации учреждений и предприятий, средств автоматизации разработки программных систем (CASE-средств).

Крупнейшим партнером фирмы Sequent является компания Oracle, доминирующая на мировом рынке реляционных баз данных. Сегодня системы Sequent являются основной платформой для установки системы управления финансами Oracle Financials.

В январе 1992 года фирма Sequent заключила стратегическое соглашение с компанией Texas Instruments о разработке CASE-систем для своих машин.

Среди пользователей систем Sequent — промышленные и страховые компании, банки, коммуникационные компании, правительственные организации США, органы здравоохранения, университеты и исследовательские центры.

Крупнейшая в мире страховая компания Lloyd использует системы Sequent для выполнения сложных программ учета и оценки рисков страховок.

Транснациональная нефтехимическая компания British Petroleum выбрала системы Sequent и СУБД ORACLE в качестве базы для создания фирменной общеевропейской сети на базе протокола X.25, которая позволит координировать работу отделений компании в 16 странах.

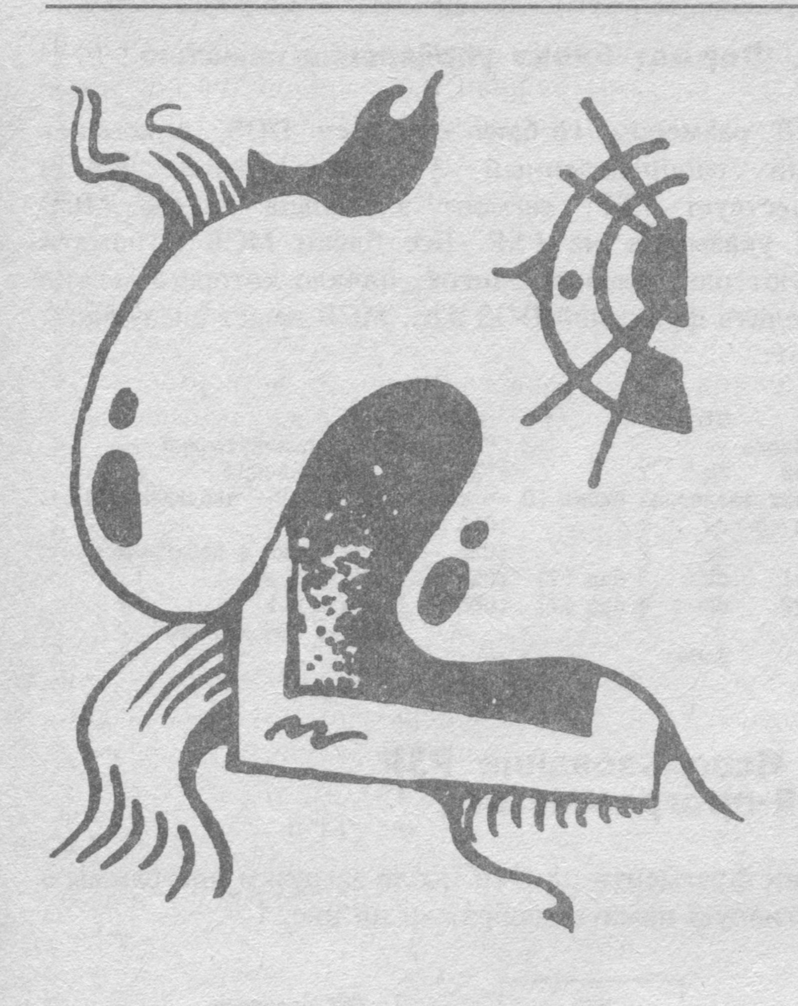
Компания Volkswagen, занимающая четвертое место среди ведущих мировых производителей автомобилей, установила систему Sequent Symmetry 2000/700 на своем заводе в Пуэбло (Мексика) и реализовала на ней АСУ предприятия. Завод выпускает четыре модели легковых автомобилей и является крупнейшим в стране.

Американская телефонная компания US WEST NewVector использует системы Sequent в качестве серверов своей сети пакетной передачи данных и радиотелефонной связи для управления коммутаторами и учета оплаты коммуникационных услуг. Сеть обслуживает более 370 000 абонентов в 14 штатах США.

Системы Sequent, благодаря их мощности, открытости, уникальным возможностям наращивания и умеренной цене, представляются весьма привлекательными для тех пользователей из СНГ, которые имеют желание решать насущные и весьма масштабные задачи автоматизации и интеграции биржевой, банковской и страховой деятельности, создания информационных систем и систем связи, а также задачи АСУ, но не располагают лишними деньгами.

Архитектура систем Sequent избавляет пользователя от необходимости поиска ЭВМ для оптимального решения вышеназванных задач, поскольку эти машины могут “расти“ вместе с прикладной системой, причем именно так, как это нужно пользователю. В отличие от многих своих конкурентов, фирма не навязывает конечному пользователю практически ни одного программного или аппаратного решения, в наибольшей степени реализуя тем самым принцип открытости и сообразуясь с девизом свободного рынка — “покупатель всегда прав“.

*Л.Павлов*



*Если вы накоротке с ассемблером, то вам наверняка знакомы понятия TSR и PSP. Однако, то ли из страха, а скорее — от недостатка достоверной информации, программисты редко используют пространство PSP в своих резидентных программах. А ведь для короткой и изящной TSR-программы "лишние" 256 байт памяти — весьма лакомый кусочек. Проглотить же этот кусочек вам поможет*

# Использование PSP в TSR-программах

**PSP** *(Program Segment Prefix)* — префикс программного сегмента.
**TSR** *(Terminate and Stay Resident)* — завершить и остаться резидентом.
**MCB** *(Memory Control Block)* — блок управления памятью.

Что побудило меня написать данную статью? Не так давно я прочел работу "группы программистов 2В" В.А.Вегнера и В.В.Серегина: "Способ создания TSR-программ без PSP", из которой следует, что все авторские права по использованию PSP в TSR-программах принадлежат "группе 2В". Признаться, такое заявление меня несколько озадачило, поскольку я, как выясняется, вот уже 2 года "нарушаю права" "группы 2В", используя PSP в своих TSR-программах.

"Группой 2В" разработан оригинальный алгоритм создания TSR-программ без PSP:

- освобождается ENVIRONMENT и корректируется размер памяти, занимаемой программой; при этом DOS строит новый блок MCB сразу за концом будущей TSR-программы;
- в теле самой программы сразу за MCB средствами DOS строится новый PSP, у которого корректируется адрес родительского PSP (копируется со старого);
- новый PSP указывается DOS в качестве текущего;
- корректируется указатель на собственника памяти в новом MCB на сегментный адрес нового PSP;
- резидентная часть программы пересылается по адресу сегмента старого PSP со смещением 0;
- далее программа завершается обычным образом по функции DOS 04ch или по INT 20h.

Мною разработан иной алгоритм использования PSP TSR-программами, который был успешно применен при разработке резидентных программ — драйверов



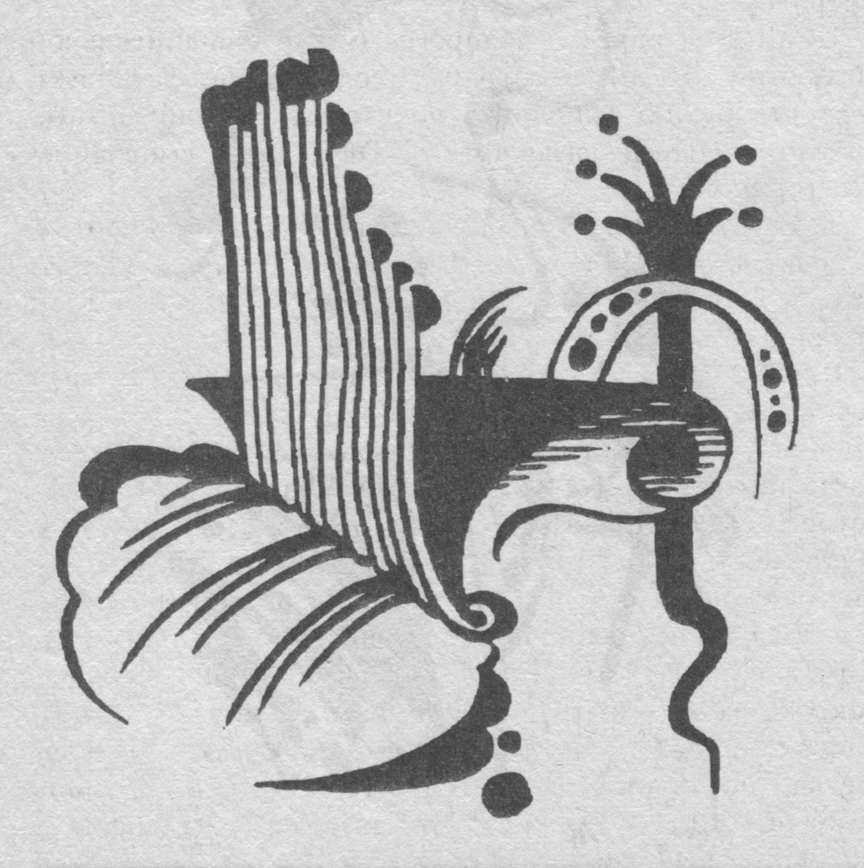
PU_KBG, PU_KBGM, PU_DISKM, PU_VEGAM, PU_TIMEM, PU_INPRN и многих других.

# 1. Основные понятия

## 1.1. Формат префикса программного сегмента

PSP размером 256 байт строится DOS при запуске программы, всегда предшествует ей и имеет следующий формат:

```
PSP?        Struc
PSP_int_20  db   0cdh,020h  ;00h -- INT 20 для завершения
;                                   программы
PSP_top     dw   ?          ;02h -- N 1-го свободного параграфа
PSP_rsv1    db   6  dup(?)  ;04h -- Р е з е р в
; Адрес завершения программы -- копия вектора 22h:
PSP_tm_ip   dw   ?          ;0ah
PSP_tm_cs   dw   ?          ;0ch
; Адрес выхода по Ctrl-Break -- копия вектора 23h:
PSP_cb_ip   dw   ?          ;0eh
PSP_cb_cs   dw   ?          ;10h
; Адрес выхода по фатальной ошибке -- копия вектора 24h:
PSP_er_ip   dw   ?          ;12h
PSP_er_cs   dw   ?          ;14h
; Сегментный адрес родительского PSP
PSP_ow_cs   dw   ?          ;16h
; File Handle Table -- 20 индексов для системной табл. файлов
PSP_fht     db   20 dup(?)  ;18h -- File Handle Table
; Сегментный адрес среды процесса (ENVIRONMENT)
PSP_env     dw   ?          ;2ch
PSP_rsv2    db   34 dup(?)  ;2eh -- Р е з е р в
PSP_dos     db   0cdh,021h  ;50h -- INT 21 -- обр. к ф. DOS
PSP_rsv3    db   3 dup(?)   ;52h --  Р е з е р в
PSP_efcb1   db   7  dup(?)  ;55h -- расширение FCB1
PSP_fcb1    db   16 dup(?)  ;5ch -- Неоткрытый FCB1
PSP_fcb2    db   16 dup(?)  ;6ch -- Неоткрытый FCB2
PSP_rsv4    db   4  dup(?)  ;7ch -- Р е з е р в
; Область DTA по умолчанию и:
PSP_lparm   db   ?          ;80h -- Длина поля параметров
PSP_parm    db   127 dup(?) ;81h -- Поле параметров
PSP?        Ends
```



## 1.2. Формат блока управления памятью

MCB размером 16 байт строится DOS, описывает каждый распределенный участок памяти, всегда предшествует PSP; сегмент владельца такого MCB всегда указывает на PSP. Все блоки MCB в памяти образуют односвязный список, начало которого можно определить функцией DOS 52h. MCB имеет следующий формат:

```
MCB?        Struc
; Тип блока --                  "M" (4dh) -- промежуточный
MCB_type    db   ?             ;"Z" (4ah) -- конечный
; Сегмент владельца блока (0 -- свободный блок -- владелец DOS):
MCB_own_cs  dw   ?             ;01h -- CS
MCB_Lp      dw   ?             ;03h -- длина блока в параграфах
MCB_rsv1    db   3 dup (?)     ;05h -- Р е з е р в
MCB_rsv2    db   8 dup (?)     ;08h -- Р е з е р в
;                                      (обычно имя хозяина)
MCB?        Ends
```

# 2. Использование PSP TSR-программами

План фрагмента памяти после загрузки программы в оперативную память изображен на рис. 1.

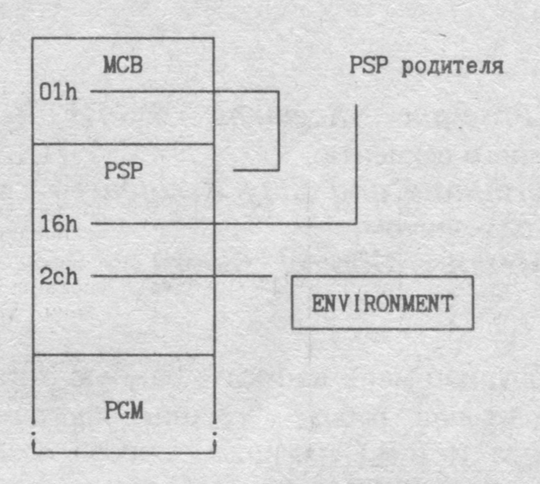



Рис. 1. План памяти при загрузке программы

Если TSR прямо или косвенно не использует область ENVIRONMENT, то ее надо освободить.

TSR-программы, использующие весь PSP, называют TSR-программами без PSP. Ниже дается алгоритм построения программ без PSP (или с частичным его использованием), разработанный автором. Сущность его состоит в перемещении программы в область PSP ПОСЛЕ ЗАВЕРШЕНИЯ функции DOS 31h или INT 27h.

Хорошо известный у нас в стране драйвер “800“ использует только нижнюю половину PSP.

## 2.1. Алгоритм инициализации TSR-программ с использованием PSP

1. Определить конец резидентной части программы (EndRes), выровненный по границе параграфа.
2. Запомнить 16 байт резидентной части программы по смещению EndRes-Delta в специально отведенном

буфере. Delta — величина перемещения программы (кратна 16). Для TSR без PSP Delta = 100h.
В эту область DOS поместит последний MCB-блок после резидентного завершения программы (рис. 2).
3. Освободить блок среды, если это необходимо.
4. Переопределить указатель выхода PSP_tm (смещение 0ah) указателем на программу-аппендикс выхода в DOS (п. 2.2), предварительно запомнив вектор int 22h в DD-переменной.
5. Вычислить длину резидентной части программы в параграфах с учетом перемещения в PSP:
   LTSR = (EndRes-Delta)/16
   Запомнить, как минимум, регистры SS и SP.
6. Завершить программу по функции DOS 31h или INT 27h.

### 2.2. Алгоритм программы-аппендикса выхода в DOS

1. Восстановить, как минимум, регистры SS и SP.
2. Переместить резидентную часть программы в PSP.
3. Восстановить 16 байт программы по смещению EndRes-2*Delta от нового начала программы из буфера.

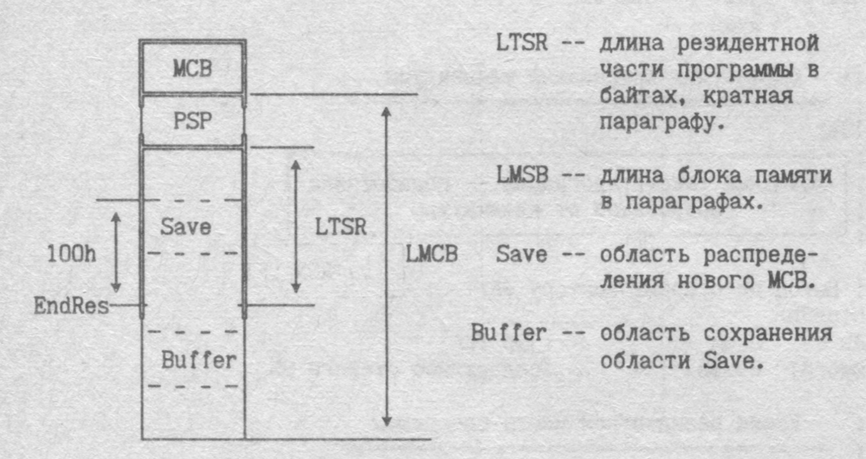



Рис. 2. План памяти на входе программы-аппендикса выхода в DOS (Delta=100h)

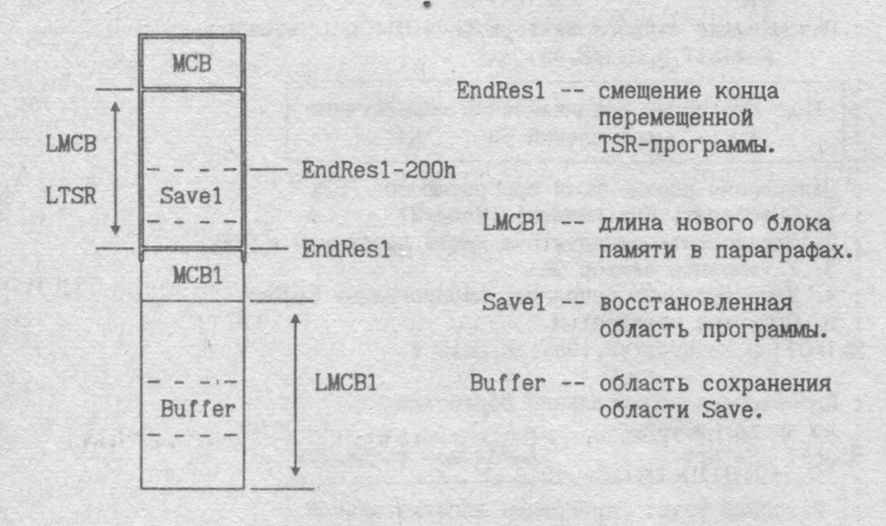



Рис. 3. План памяти на выходе программы-аппендикса выхода в DOS (Delta=100h)

4. Установить требуемые вектора прерываний, если это необходимо. При установке векторов необходимо учесть новое значение сегмента TSR = CS-Delta/16.
5. Осуществить длинный переход по запомненному вектору 22h.

## 3. Реализация алгоритма

Реализован только для COM-программ одной макрокомандой EXir0? и одним объектным модулем PU_Exir0. Макрокоманда Exir0? осуществляет построение блока управления завершением TSR-программы и вызов модуля PU_Exir0, реализующего вышеуказанный алгоритм.

### 3.1 Формат макрокоманды Exir0?

```
Exir0?  [TPSP],EndRes,[VList],[Keep],[UCall]
```

TPSP — длина в параграфах неизменяемой части PSP — определяет адрес начала перемещенной TSR-программы. Для TSR-программ без PSP TPSP=0. По умолчанию = 0.

EndRes — смещение конца резидентной части программы (выравнивание на параграф не обязательно).

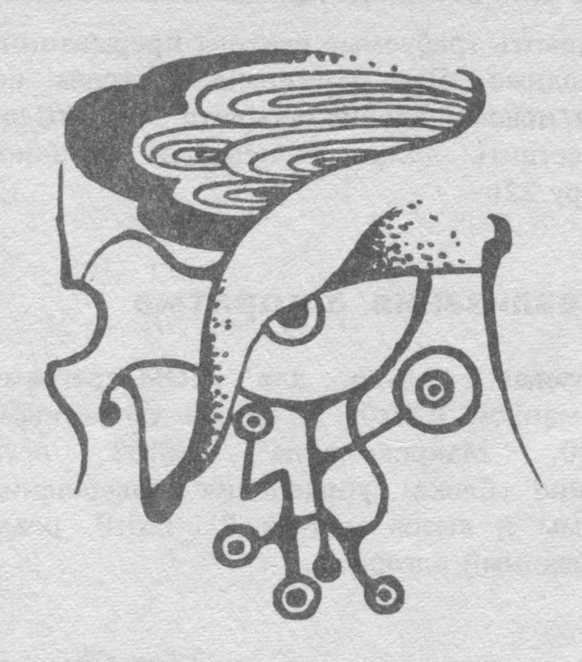
VList — список устанавливаемых векторов в формате:
<<N1,Off1>,<N2,Off2>,...>
где N1, N2, ... — номера векторов; Off1, Off2, ... — смещения программ-обработчиков прерываний. По умолчанию список пуст.

Keep — =Y/[N] — предписывает сохранить/освободить блок среды (ENVIRONMENT). По умолчанию Keep=N.

UCall — адрес подпрограммы дополнительной обработки. UCall необходим, если кроме установки векторов нужно выполнить некоторые дополнительные операции перед выходом в DOS. По умолчанию отсутствует.

Таким образом, процесс создания TSR-программы, использующей PSP или ее часть, сводится к выдаче макрокоманды Exir0? с необходимыми операндами в исходном тексте программы.

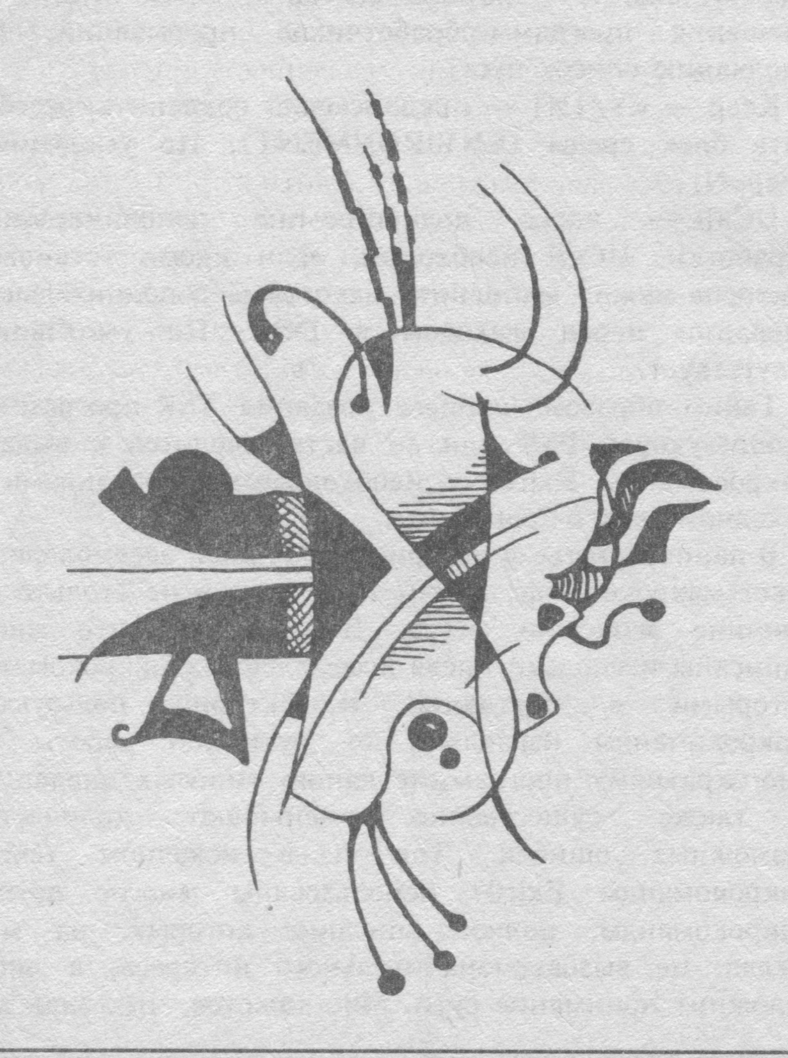
В данной статье я не привожу полный ассемблерный текст макрокоманды Exir0? не только и не столько по причине экономии места. Дело в том, что мною написаны несколько тысяч ассемблерных макрокоманд, которыми я постоянно и активно пользуюсь. Макрокоманды избавляют от рутинной работы по многократному программированию типовых операций, а также существенно уменьшают количество возможных ошибок. Так и в исходном тексте макрокоманды Exir0? использованы многие другие макрокоманды, полное описание которых, на мой взгляд, не вызовет значительного интереса, а лишь усложнит понимание сути. Мне кажется, что, зная ал-



горитм, читатели сами сумеют его запрограммировать, что также поможет им лучше разобраться с использованием PSP TSR-программами.

### 3.2 Пример построения фрагментов TSR-программы без PSP

В приведенном ниже примере программы использованы макрокоманды выполнения типовых операций. Смысл их интуитивно понятен из текста программы. Для облегчения восприятия все макрокоманды снабжены отличительной особенностью — их имена заканчиваются вопросительным знаком.



```
        page  88,132           ;Размер страницы

; Пример построения TSR-программы без PSP
; Программист  Панков Ю.И.

; Дата создания:    4. 3.91
; Дата редакции и N версии:
DAT_@@@ equ    <04.03.91>
VER_@@@ equ    <V 1.0>

Nam_@@@ equ    <"PU_PSPRS">
name    PU_PSPRS

out? <.     ╔════════════════╗>
out? <.     ║ P U _ P S P R S ║>
out? <.     ║                ║>
out? <.     ║ >,%VER_@@@,<   >,%DAT_@@@,< ║>
out? <.     ╚════════════════╝>

;  Подключение сервиса
        include MACRO_OO.INC      ;общего назначения
        include MACRO_IO.INC      ;функции BIOS
        include MACRO_01.INC      ;дополнительные
        include MACROTBL.INC      ;генерация таблиц
        include MACRODOS.INC      ;функции DOS
        include MACROKBD.INC      ;функции BIOS
;  Управление листингом МАКРО
;       .lall               ;полный
;       .xall               ;неполный
        .sall               ;без расширений макро

        .model  small
        .code               ;начало кодового сегмента _Text
        assume  ds:_text
        org     100h        ;для COM-программ
Start:  jmp     Start1
        even

;    Обработчик прерываний клавиатуры
 ══════════════════════════════════════
I9h:
;
;  ┌────────────────────────────────────────┐
;  │ Исходный текст  программы -- обработчика │
;  │         прерываний от клавиатуры         │
;  └────────────────────────────────────────┘
;

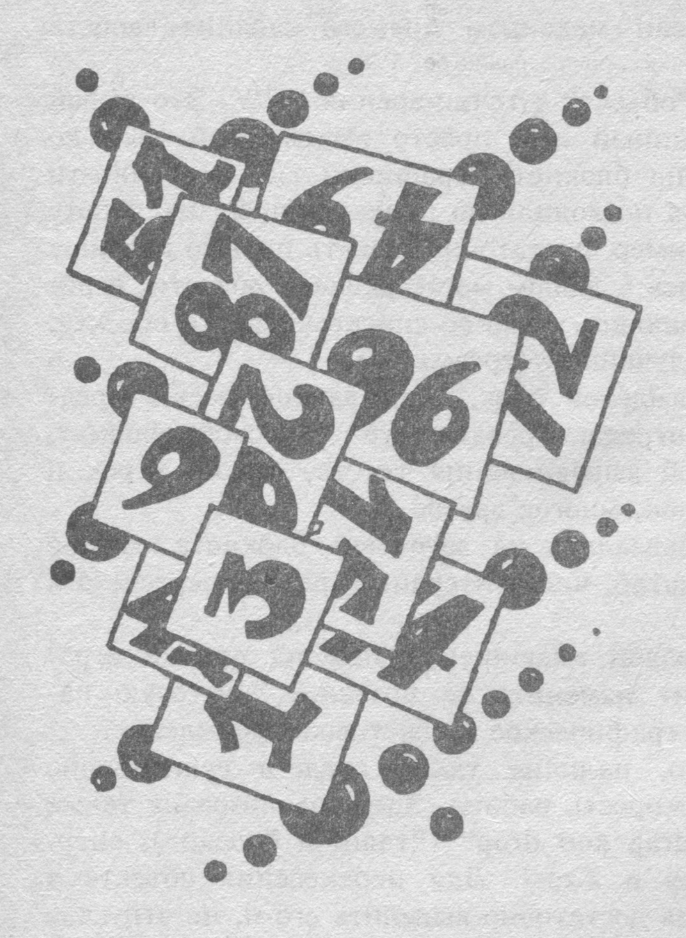
; Выход по старому вектору 9h:
Bios9h:
db      0eah                ;jmp far
Dword?  Old9h               ;содержимое старого 9h

;    Конец резидентной части программы
 ══════════════════════════════════════
;
EndRes equu     $

;Запуск программы
Start1:
        cld
; Запоминание старого вектора 9h в Old9h (сохранить es):
        GetInt? 9,OldKB,es
;
;  ┌─────────────────────────────────────────┐
;  │ Тело программы инициализации обработчика │
;  │            прерываний 9h                 │
;  └─────────────────────────────────────────┘
;
; Завершение резидентной программы без PSP:
; 1. Освободить Environment (Keep=N)
; 2. Переместить резидентную часть программы в PSP
; 3. Установить вектор 9h
; 4. Дополнительно выполнить подпрограмму SetDrv
; 5. Остаться резидентом
ExirO?  0,EndRes,<<9,I9h>>,N,SetDrv

; Программа дополнительной обработки
; на входе: ds=es=cs
Block?  SetDrv              ;заголовок процедуры
;
;  ┌────────────────────────────────────────┐
;  │ Исходный текст  программы дополнительной │
;  │              обработки                   │
;  └────────────────────────────────────────┘
;
Bend?   SetDrv              ;конец процедуры
        END     Start
```

*Ю.Панков*



*Электронные таблицы — самый распространенный в мире класс программного обеспечения. Этому есть две причины: первая — легкость освоения, вторая — эффективность использования.*

# Quattro Pro for Windows заявляет о себе

Действительно, владение начальными навыками работы на компьютере входит в необходимый минимум знаний любого человека, работающего в сфере менеджмента. А электронная таблица способна быть единственным и достаточным средством общения с "персоналкой" практически всех сотрудников обычной организации. Подведение итогов, подготовка отчетов, написание писем, просто вычисления — вся повседневная работа может выполняться с помощью электронной таблицы на порядок быстрее, нежели без нее.

В настоящее время рынок электронных таблиц превратился в арену сражения гигантов — всемирно известные американские корпорации Borland, Lotus, Microsoft пытаются укрепить свое влияние среди широких масс пользователей. Новые версии Quattro Pro, 1-2-3, Excel появляются с периодичностью раз в полгода, открывая все новые и новые горизонты.

Естественно, с появлением графической среды Windows большой интерес возник к программным продуктам, способным использовать все ее возможности. Первой электронной таблицей для Windows стала Excel (фирма Microsoft), затем появилась Lotus for Windows. И наконец, объявление корпорации Borland о выпуске Quattro Pro for Windows состоялось летом 1992 года. Отныне Borland будет выпускать этот пакет параллельно с Quattro Pro 4.0 — версией для DOS. Сравнению двух борландовских электронных таблиц и посвящена эта статья.

## Первое включение

Итак, после установки Quattro Pro for Windows (QPW) появилось желание запустить программу. Два щелчка левой клавишей мыши, и... довольно долгое мигание лампы-индикатора жесткого диска, в процессе которого вспоминается, что Quattro Pro for Windows занимает на винчестере свыше 10 Мбайт и требует по крайней мере 4 Мбайта оперативной памяти.

От этих мыслей отвлекает появление на экране монитора рабочей области таблицы. Сначала большое значение играет привычка — ведь все версии Quattro Pro для DOS сохраняли преемственность интерфейса, а здесь абсолютно другой вид рабочей области. Прежнее меню казалось достаточно удобным, а что нам уготовано здесь? Сразу стараешься найти "что не так", какие прежние возможности стали недоступными. Однако вскоре становится ясно, что потерь нет — можно по-прежнему работать с системой команд Quattro Pro 4.0 и Lotus 1-2-3. Кнопочная панель мыши может быть отредактирована в соответствии с запросами пользователя. Последний штрих — подбор привычной цветовой гаммы. Эта процедура реализуется легко, результаты превосходны: естественно, от Windows ничего другого и не ожидали.

Теперь можно посмотреть и новые возможности — если не понравится, отступим в старое окружение.

## Первый взгляд на интерфейс

В основу организации рабочей области QPW положена новая метафора — метафора блокнота. Теперь каждый файл представляет собой трехмерную таблицу размером 8192x256x256.

Большинство людей в повседневной жизни используют записные книжки, телефонные справочники и другие хранилища информации. И многие находят это

привычным и удобным. Что может быть проще? Достаточно зажать пальцем нужную букву на правом обрезе страницы, и ваша записная книжка откроет нужный перечень имен и телефонов. Впрочем, почему обязательно "имен и телефонов"? Листы книжки можно также использовать для записи колонок цифр или для зарисовки графиков. Идя навстречу устоявшимся пристрастиям пользователей, корпорация Borland предложила аналогичную схему работы в пакете Quattro Pro for Windows.

Подобно записной книжке, при работе с QPW информация заносится на страницы файла-блокнота. Каждая страница — это обычная двумерная таблица. Третье измерение появляется из-за того, что страницы можно листать. В нижней части рабочей области QPW (над строкой состояния) находится список с названиями страниц. Для перелистывания достаточно спозиционировать указатель мыши на ярлык нужной страницы и нажать левую клавишу.

Благодаря реализации такой технологии эффективность работы существенно возрастает. Например, бухгалтер теперь получает возможность хранить годовую отчетность на листах с именами "январь", "февраль", "март" и далее, а также создавать трехмерные, "сквозные" через все листы формулы и графики.

Различные атрибуты, такие как цветовая гамма, шрифтовое и графическое оформление, определяются для отдельной страницы или для всего блокнота. QPW имеет большую гибкость при работе с файлами: пользователь может открывать несколько раз одну таблицу, создавать многооконные приложения, разрезать окна с синхронизацией и без оной, а также фиксировать заголовки.

## Прямое задание свойств

Всем известна любовь руководителей корпорации Borland к миру объектов. Работа с готовыми формами действительно очень эффективна и удобна. Поэтому наряду с традиционным меню в Quattro Pro for Windows реализован механизм прямого задания свойств объектов.

Что такое "объект" с точки зрения QPW? Это любой блок, именованный или просто выделенный мышью, любая страница блокнота, график и т.д. Особенности среды Windows позволили до предела упростить работу с ними. Например, достаточно нажать правую клавишу мыши, находясь в любом месте рабочей области, и появится всплывающее меню со списком свойств объекта, который мы случайно потревожили.

Если был выделен блок, всплывающее меню предложит вам отредактировать его числовой формат, шрифт, способ выравнивания текста, высоту строк и ширину колонок, многое другое.

Если мы оказались на заголовке блокнота, можно изменить масштаб, число итераций, режим пересчета и т.д.

Щелчок правой клавишей мыши на ярлыке страницы позволит изменить ее название, цветовую палитру, другое графическое и текстовое оформление.

Несомненно, наличие таких средств существенно увеличивает скорость работы. Здесь реализована также технология "drag and drop" ("тащи и бросай"), впервые введенная в Excel. Для перенесения объекта в таблице теперь достаточно выделить его и, не отпуская левой клавиши мыши, перенести блок в новое место. В числе других интересных особенностей редактирования стоит упомянуть непосредственное изменение высоты строк и ширины колонок, а также выделение всей строки или всей колонки щелчком мыши, спозиционированной на координате строки или колонки.

## Графические возможности

С самых ранних версий электронная таблица Quattro Pro славилась своей графикой. Что можно предложить дополнительно к широкому выбору графиков Quattro Pro 4.0, что можно добавить в графический редактор, как упростить работу с аналитическими графами? Можно ли улучшить потребительские свойства Quattro Pro 4.0, программного продукта, по графическим возможностям конкурирующего с издательскими системами?

Пока возникают эти вопросы, мышь позиционируется на пункте "Graph". Щелчок левой клавишей, и — сначала среди загашенных указателей одиноко мерцает слово "New". Однако затем все встает на свои места: как матрешки, один за другим начинают раскрываться знакомые по DOS-версиям списки выбора.

QPW поддерживает все типы графиков Quattro Pro 4.0, за исключением пузырьковых диаграмм. Имеется возможность строить двумерные линейные, вертикальные и горизонтальные столбцевые, стопочные, круговые, площадные, процентные и биржевые диаграммы, а также X-Y графики.

Из множества трехмерных графиков стоит выделить столбцовые, ленточные, ступенчатые и графики-поверхности второго порядка в трехмерном пространстве.

Для большинства типов графиков можно применять операцию поворота для обзора "сверху". Кроме того, появились новые типы диаграмм — Combo, благодаря чему несколько серий значений можно рассматривать как отдельные круговые диаграммы, как линейный график на фоне площадной диаграммы и т.п.

Графики автоматически запоминаются на специальной странице в конце блокнота. Поэтому для повторного вызова графика достаточно щелкнуть левой клавишей мыши, стоя на иконке с изображением диаграммы.

При открытии листа с графиком происходит автоматическая загрузка графического редактора. Его возможности весьма широки и включают в себя:
- полный набор примитивов (прямоугольники, эллипсы, ломаные, стрелки и т.д.);
- масштабирование и позиционирование;
- выравнивание и сеточное притяжение объектов.

Безусловно, очень полезна возможность импортировать прямо в таблицу графические изображения, хранящиеся в любом поддерживаемом Windows формате.

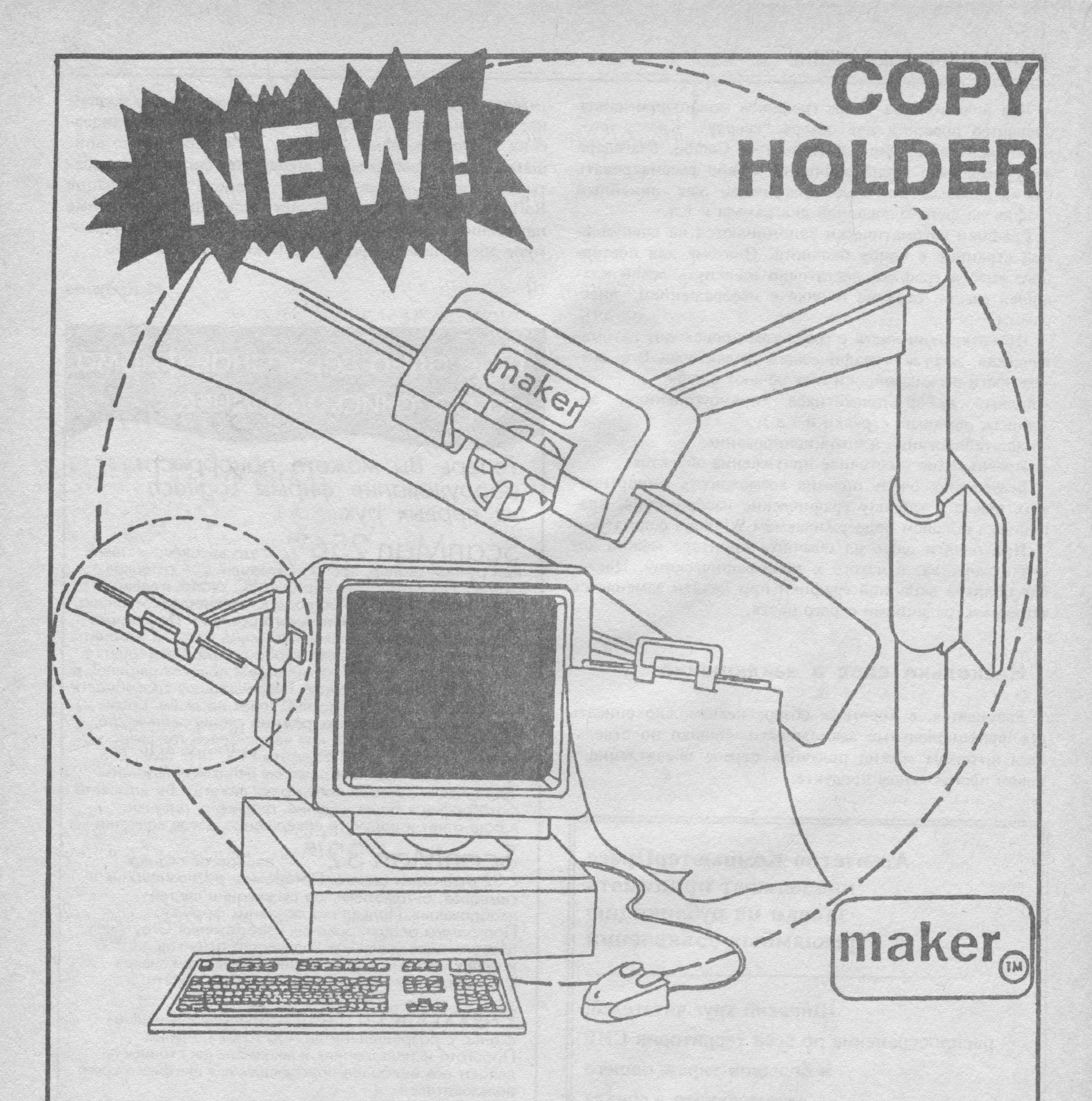
При печати даже на обычном принтере можно достичь качества, близкого к полиграфическому. Цветовая палитра экранной графики при печати заменяется теневыми градациями серого цвета.

## Несколько слов в заключение

Разумеется, в коротком обзоре невозможно описать все функциональные возможности. Однако по отдельным штрихам можно получить первое впечатление о новом программном продукте.

Итак, Quattro Pro for Windows — мощная электронная таблица, унаследовавшая лучшие черты борландовских бизнес-программ. Видимо, QPW гармонично впишется в ряд программных продуктов фирмы. Разрабатывая новые продукты "под Windows", корпорация Borland еще раз подтверждает свои самые серьезные намерения играть роль законодателя моды на программное обеспечение персональных компьютеров.

*И.Бодунов*




**COPY HOLDER** закрепите на мониторе, кассовом аппарате и т.п. (крепление в комплекте). Вставьте лист с текстом, найдите удобный для глаз угол наклона.

**COPY HOLDER** сохранит Ваше зрение и время, освободит рабочее место.

**COPY HOLDER — ЭТО КОМФОРТНО И ПРЕСТИЖНО!**

Приглашаем дилеров к сотрудничеству!

Тел. (095) 400 7937, 312 4903, 559 7126, 335 9377

*Трудные времена, о которых мы все так долго мечтали, увы, все еще продолжаются. Но право свободного выбора гарантировано нам на долгие времена, так что, уважаемые научные работники, если вы еще не нашли своего избранника, то читайте дальше статью про суперкалькуляторы, начало которой вы найдете в предыдущем номере КомпьютерПресс.*

# Это вам, романтики, это вам, ученые

## TKSOLVER

1. Объем памяти: 290 Кбайт.
2. Типы графиков: сплошная линия, столбчатая и круговая диаграммы, отрезки, соединяющие точки (при работе с электронной таблицей).
3. Логарифмический масштаб на графиках: есть.
4. Видеоадаптеры: EGA, CGA, Hercules.
5. Печать: только PrtSc ("печать с экрана") и команды DOS Graphics.
6. Математический сопроцессор: используется, но необязателен; если имеется в системе, то обнаруживается автоматически.
7. Функции: стандартный набор, гамма, Бессель; дополнительно большое число функций реализуется через систему электронных таблиц — это, в частности, интегрирование по методу Симпсона, численное дифференцирование, БПФ.
8. Поддержка матричной алгебры: реализуется посредством отдельно загружаемых модулей и включает умножение, инвертирование, вычисление определителей и собственных значений.
9. Решение уравнений: как методом подстановки, так и методом Ньютона.
10. Комплексные числа: ограниченная поддержка (довольно богатые возможности, но для работы с комплексными величинами требуется особая форма записи).
11. Условные переходы: есть, очень развиты.
12. Возможность создавать собственные функции: есть.
13. Возможность обращения к внешним программам, написанным на других языках программирования: нет.
14. Предупреждает о необходимости сохранить результат: нет (требует только подтверждения решения о выходе).
15. Выход в DOS: нет.

TKSOLVER является наиболее мощным из пакетов универсального назначения. К этому классу можно отнести также EUREKA и MATHCAD. В сравнении с TKSOLVER они выглядят значительно слабее. Помимо решения систем уравнений (методом подстановки или методом Ньютона) пакет располагает развитым языком программирования и поставляется с богатой библиотекой электронных таблиц с функциями, которые можно использовать в различных целях: быстрое преобразование Фурье, численное дифференцирование, аналитическое дифференцирование и даже решение систем дифференциальных уравнений методом Рунге-Кутта.

В качестве основного варианта организации работы пользователя (так называемые метафоры) TKSOLVER использует электронную таблицу, занимающую на экране одно окно. Всего предусмотрено до девяти основных окон, но каждое может иметь так называемые подтаблицы. Имеются две основные таблицы — таблица правил и таблица переменных. Если, набрав какое-то уравнение, вы хотите, чтобы оно рассматривалось как одно из правил, то пакет автоматически дополняет список переменных переменными вновь введенного уравнения. То, что переменные размещаются в одном списке — очень удобно при их просмотре на экране. Определения процедур и функций хранятся в другой таблице; есть отдельная таблица для матриц. На экране одновременно может быть не более двух окон, а ввод символа "=" в командном режиме позволяет получить список всех окон.

Пакет довольно успешно обращается с синтаксическими ошибками и ошибками, возникающими вследствие несовместности систем уравнений. Уравнения, в которых обнаружено что-то не то, помечаются на полях символом "»". Перемещение курсора на такой символ приводит к появлению сообщения, описывающего ошибку.

Пакет, несомненно, очень мощен, что в ряде случаев может оказаться даже лишним. Конечно, приятно ворваться в соседнюю комнату к сослуживцам и порадовать их: "Ребята, я нашел такую программу! Она может решать дифуры по Рунге-Кутту!" Но видимо, если вам действительно потребуется этот метод, то вы быстренько напишете программу для какого-нибудь компилятора и не станете пользоваться достаточно медленным интерпретатором типа TKSOLVER. Еще одно неприятное следствие высокой мощности пакета — более сложный пользовательский интерфейс и немалые усилия, необходимые для его настройки, способной облегчить доступ к часто используемым функциям. В EUREKA и MATHCAD численное дифференцирование реализовано как встроенное средство, а в TKSOLVER для регулярного использования этой функции, ее надо перенести из библиотеки в так называемую "модель". Вывод прост: если такого рода программа нужна для каждодневной, регулярной работы (например, решения систем уравнений), то, наверное, стоит потратить усилия на мастерское овладение ею. Вы скоро поймете (если не вдаваться в подробности), что это именно то, что нужно. Но если пакет нужен лишь изредка, то лучше обратиться к чему-нибудь типа MATHCAD.

Уникальная черта TKSOLVER — сочетание методов решения систем уравнений с другими функциональными модулями. Если нужно, например, графически изобразить параметрическое уравнение в случае, когда зависимости переменных от параметра не явны, то TKSOLVER справится с таким заданием довольно легко, в то время как для других пакетов это сложная, а иногда и просто невыполнимая задача. Платой за такие возможности как всегда оказывается более сложный (по сравнению с другими пакетами) сам процесс построения параметрических уравнений.

Известны две усеченные версии TKSOLVER:

- вариант для студентов, по существу, основная версия, но без развитых библиотек функций для решения дифференциальных уравнений, численного интегрирования и т.п.);
- демонстрационная версия (поставляется бесплатно). Позволяет решать систему из 24 скалярных уравнений с числом переменных не более 32, имеет только встроенные функции, не работает с массивами и не имеет встроенного языка программирования.

Имеются модификации TKSOLVER для компьютеров Macintosh, Apple II, TRS-80 и машин, не совместимых с MS-DOS, — например, DEC Rainbow.

## MATHCAD

1. Объем памяти: 340 Кбайт.
2. Типы графиков: сплошная линия, в виде несоединенных точек, столбчатая диаграмма.
3. Логарифмический масштаб на графиках: есть.
4. Видеоадаптеры: EGA, CGA, Hercules.
5. Печать: матричные и лазерные принтеры, графопостроители (но не работает с PostScript).
6. Математический сопроцессор: используется, но необязателен; если имеется в системе, то обнаруживается автоматически.
7. Функции: стандартный набор, статистика, Бессель, БПФ, линейная регрессия, линейная интерполяция, кубическая экстраполяция, вычисление гамма-функции, численное интегрирование и дифференцирование.
8. Поддержка матричной алгебры: есть, умножение, инвертирование и вычисление определителей.
9. Решение уравнений: есть.
10. Комплексные числа: полная поддержка.
11. Условные переходы: очень ограниченные возможности (только две функции: "если (условие, v1, v2)" возвращает v1...v2, если условие верно.
12. Возможность создавать собственные функции: есть.
13. Возможность обращения к внешним программам, написанным на других языках программирования: нет.
14. Предупреждает о необходимости сохранить результат: да.
15. Выход в DOS: да.

Если не принимать во внимание низкое быстродействие, MATHCAD можно считать идеалом. Не то чтобы к нему не было никаких претензий, но за исключением обработки символьных выражений, в нем есть все вообразимые функции, не только стандартные, включая статистические, но даже функции, позволяющие решать, например, задачи линейной корреляции или кубического экстраполирования. В части решения систем уравнений пакет располагает методом Ньютона. Есть в нем и метод быстрого преобразования Фурье, и интегрирование (методом Симпсона), и дифференцирование. А на принтер выводятся даже настоящие математические символы, например, символ суммы с индексами для обозначения операции суммирования. Ну и если вам просто нужно что-нибудь не слишком сложное посчитать, то тут даже документация не потребуется.

Ахиллесовой пятой пакета является его быстродействие. Здесь срабатывают два фактора. Во-первых, пакет теряет в быстродействии на вычислительных частях задач. Кроме того, по умолчанию он вычисляет формулы в таком темпе, что при прокрутке на экране можно увидеть сообщение "ждите". Можно, конечно, отключить режим автоматического пересчета, но прокрутка все равно будет идти медленно, и тут сказывается второй фактор — компьютер работает в графическом режиме. Кроме того, надо будет не забывать нажимать периодически клавишу Calc. Хуже того, перед выводом на печать вместо этой клавиши потребуется специальная команда, по которой пакет пересчитает всю математику выводимого документа.

Одним из недостатков пакета можно считать ограниченность языка. Однако хорошенько изучив векторные функции, вы сможете выпутаться из любого положения. Например, попробуйте вычислить наибольший общий делитель — хотя в руководстве сказано, что сделать это невозможно.

Пользовательский интерфейс позволяет вводить команды и при помощи функциональных клавиш, и через меню, в командной строке. MATHCAD легко рас-

познает комплексные числа — например, вы можете, если захотите, попросить его вычислить выражение вида sin(2i).

Еще несколько моментов может активно не понравиться. Например, в редакторе клавиши Del и Backspace выполняют одно и то же действие, а клавиша Inc переключает редактор в один из двух режимов вставки! Как оказалось, режима overtype вообще нет. Нет возможности и изменять функциональное назначение клавиш. Например, символ "[" используется для обозначения нижнего индекса — наверное, можно было бы найти и что-нибудь получше. Никакого удовольствия не доставляет и то обстоятельство, что для выполнения некоторых операций курсор надо переводить в определенные места экрана. Например, если вы хотите, чтобы переменная "n" стала равной 2, вы должны набрать "n:=2", но если начать набор с символов "n= ..", то немедленно появится сообщение, а символ "=" удалить, не удаляя одновременно и "n" становится невозможным, поскольку после выдачи сообщения об ошибке нажатие на клавишу "Backspace" не вызывает удаления последнего введенного символа. По какой-то причине MATHCAD индексирует векторы и матрицы с нуля. Понятно, почему это делается в языках типа ассемблера или СИ (там индексы представляют собой смещения от некоторой базы), и почему такая возможность, как вариант, предусматривается в APL, но зачем это надо было делать в данном случае? Несмотря на то, что при конфигурировании способ нумерации элементов можно изменить, все же это умолчание неудачно и довольно странно для пакета, в котором во всех остальных отношениях делается упор именно на максимальное приближение к стандартной математической нотации.

Подсистема помощи в рассматриваемом пакете в общем не плоха, но отсутствие контекстной ориентированности иногда ставит пользователя в довольно затруднительные ситуации. Например, выбирая формат записи уравнения в меню, вы получите список кодов текущих значений формата. Вполне естественно предположение, что если в этот момент обратиться к подсистеме помощи, то можно будет увидеть, что какой параметр означает. Не тут-то было! Как раз в этот момент помощь получить и нельзя. Чтобы все же выяснить назначение параметров, вы должны восстановить старые значения, получить справку и уж затем повторно обратиться к пункту Format меню.

Одной из сильных сторон MATHCAD является его способность выводить на экран или на печать символы, которых нет в стандартном наборе текстового режима. При этом по какой-то непонятной причине имеющийся набор символов греческого алфавита ограничен 17-ю прописными буквами. Почему-то нет возможности пользоваться курсивом или полужирным шрифтом, не говоря уже о какой-нибудь гарнитуре для заголовков разделов и т.п. И последнее. Конечно, в большинстве случаев применения суперкалькуляторов серьезные программные конструкции не требуются, но их отсутствие все же ограничивает возможности пакета.

Однако все эти недостатки не слишком существенны. И если у вас есть средненькая АТ-шка и немного терпения, то MATHCAD — это то, что вам нужно!

## POINT FIVE

1. Объем памяти: 215 Кбайт (версия для компьютера с сопроцессором 8087).
2. Типы графиков: сплошная кривая, несоединенные точки, столбчатая диаграмма, диаграмма из вертикально надстраиваемых столбцов.
3. Логарифмический масштаб на графиках: нет.
4. Видеоадаптеры: CGA, EGA (только в режиме эмуляции CGA), Hercules.
5. Печать: графики выводятся на принтер только через клавишу PrtSc.
6. Математический сопроцессор: имеется специальная версия для компьютеров с сопроцессором 8087.
7. Функции: стандартный набор (за исключением гиперболических тригонометрических функций), статистика, расширенная рядом дополнительных функций, функции для финансовых расчетов.
8. Поддержка матричной алгебры: есть умножение, инвертирование, вычисление определителей.
9. Решение уравнений: только системы линейных уравнений методом обратной матрицы.
10. Комплексные числа: нет.
11. Условные переходы: циклы "for", выражения "if-then-else", где после "then" или "else" могут стоять группы операторов. Нет "goto" и подпрограмм.
12. Возможность создавать собственные функции: нет.
13. Возможность обращения к внешним программам, написанным на других языках программирования: нет.
14. Предупреждает о необходимости сохранить результат: да.
15. Выход в DOS: нет.

POINT FIVE — пакет с довольно широкими функциональными возможностями. Он располагает и многими функциями, характерными для электронных таблиц. Все это, безусловно, расширяет круг пользователей POINT FIVE, однако чисто математические возможности пакета много скромнее, чем у остальных, за исключением разве что EUREKA. С точки зрения требований, предъявляемых к суперкалькуляторам, POINT FIVE имеет ряд очень серьезных недостатков: нельзя создавать пользовательские функции, решать системы уравнений, интегрировать, дифференцировать, работать с комплексными числами. Зато пакет располагает широкой гаммой дополнительных функций, не свойственных классическим суперкалькуляторам. Например, POINT FIVE имеет самую полную подсистему статистического анализа и ряд модулей для выполнения финансовых расчетов.

Пакет явно создавался под влиянием APL, откуда заимствована идея рабочих полей (workspace). Но по

сравнению с APL он создает куда как более дружелюбную пользовательскую среду, в основе которой лежит удачная идея разделения результатов и исходных данных. Сильные стороны POINT FIVE связаны также с заложенной в нем возможностью оперировать структурами данных. Пакет имеет встроенный редактор массивов, который вполне может конкурировать со многими электронными таблицами.

В пакете даже предусмотрен особый тип данных, имеющий название "таблица". Таблица — это матрица с поименованными строками и столбцами, что дает возможность обращаться к векторам-строкам и векторам-столбцам по их именам.

К сожалению, программа имеет и ряд раздражающих пользователя моментов (которые по заверениям разработчиков будут устранены в версии 2.0). Так, если вы хотите перемножить матрицы, то вам надо набрать на экране полностью довольно длинное ключевое слово MATRIXMULT (то же относится и к другим функциям). Возможности распечатки графиков ограничены клавишей PrtSc.

## MATLAB

1. Объем памяти: 270 Кбайт (может работать и при меньшем объеме свободной памяти, но тогда нельзя будет пользоваться графикой).
2. Типы графиков: сплошная кривая, столбчатая диаграмма, трехмерная поверхность и трехмерный контур. Графики могут строиться в полярной системе координат.
3. Логарифмический масштаб на графиках: есть.
4. Видеоадаптеры: CGA, Hercules, EGA.
5. Печать: Epson, PostScript, LaserJet, различные графопостроители. Главный недостаток — не может работать с 24-игольчатыми матричными принтерами типа Toshiba.
6. Математический сопроцессор: требуется.
7. Функции: стандартный набор, гамма-функция, статистика, полиноминальная аппроксимация, сплайн-функции, функции расчета различных электрических фильтров.
8. Поддержка матричной алгебры: очень развитая. Включает вычисление определителей, собственных значений и собственных векторов, различного рода методы декомпозиции (например, метод Шура).
9. Решение уравнений: есть.
10. Комплексные числа: есть.
11. Условные переходы: "if", "for", "while", подпрограммы, паузы до нажатия заданных клавиш.
12. Возможность создавать собственные функции: есть.
13. Возможность обращения к внешним программам, написанным на других языках программирования: Microsoft C, Borland Turbo C, Microsoft Fortran.
14. Предупреждает о необходимости сохранить результат: нет.
15. Выход в DOS: да.

MATLAB включен в обзор, поскольку по набору функциональных возможностей вполне может считаться суперкалькулятором. Но, на самом деле, это скорее язык, предназначенный для решения задач линейной математики и выполнения других сложных вычислений. Как уже отмечалось, версии пакета имеются и для ряда других вычислительных систем — VAX, Macintosh, Sun. Кроме того, это первый суперкалькулятор, который создавался с ориентацией на процессор 80386 и сопроцессор Weitek.

Работает пакет как в диалоговом, так и в пакетном режиме. В последнем случае (который значительно чаще используется на практике) выполняется предварительно написанная программа-сценарий. Большинство библиотечных функций, поставляемых с пакетом, реализовано в виде таких сценариев и хранятся во внешних ASCII-файлах с расширением .m. Собственным встроенным редактором пакет не располагает, поэтому .m-файлы приходится писать при помощи какого-нибудь внешнего текстового редактора, вызываемого из пакета. MATLAB загружает функции при первом обращении к ним. Хорошо это или плохо, но в одном файле можно описать одну и только одну функцию, поэтому через какое-то время на диске у вас появится несколько десятков таких .m-файлов.

Сценарии могут обращаться к огромному числу матричных и векторных функций, в том числе и таких экзотичных, как, например, функция факторизации по методу Чолески. Предусмотрены также логические и управляющие функции.

MATLAB без труда распознает и обрабатывает комплексные выражения типа sin(2i). Но более всего впечатляют графические средства пакета, в том числе 3-мерная графика. Пакет работает с большим числом принтеров. Правда, чтобы получить распечатку, вы должны сначала сохранить ее в специальном файле, а затем выйти из основной программы пакета и запустить специальную программу печати. Но зато графика на бумаге получается прекрасно, особенно на лазерных принтерах. Надо отметить, что если принтер располагает достаточными возможностями, то MATLAB может обеспечить его работу с разрешением 300 точек/дюйм.

MATLAB сравнительно легко осваивается и достаточно прост в использовании. И если ваш конек — матрицы, а на столе стоит 386-й компьютер с Weitek, то дальше читать эту статью не обязательно, — осваивайте MATLAB. Если же нет, то разумнее будет прибегнуть к GAUSS, обладающему более мощными средствами программирования, но требующему больше времени на изучение и освоение.

*Г.Берг*

*(Окончание следует)*



*Мы продолжаем публикацию фрагментов из книги Д.Рогаткина и А.Федорова "Turbo Pascal в среде Windows". Предыдущие главы были опубликовны в 6, 8 и 9 выпусках КомпьютерПресс.*

# TURBO PASCAL в среде WINDOWS

## Введение В GDI

Любая Windows-программа осуществляет вывод информации на то или иное устройство с помощью функций GDI (Graphics Device Interface) — графического интерфейса с устройствами. Сама среда Windows также использует функции GDI для отображения различных элементов экранного интерфейса — окон, панелей диалога, меню, иконок и т.п. Функции, реализованные в GDI, *аппаратно-независимы*, — они, как и сама программа, выводящая информацию, взаимодействуют не непосредственно с устройством, а со специальной программой, **драйвером этого устройства**. При использовании видеоадаптера EGA подключается драйвер этого устройства, видеоадаптера VGA — драйвер VGA и т.п. То же относится к различным моделям принтеров, плоттеров и другой периферии. Использование драйверов значительно упрощает смену конфигурации периферийного оборудования. При смене модели видеоадаптера, принтера или плоттера не потребуется переписывать программы — достаточно просто подключить к среде соответствующий драйвер, который и будет преобразовывать результаты выполнения аппаратно-независимых функций GDI в команды конкретного устройства.

Драйверы большого числа устройств поставляются вместе со средой и подключаются при начальной установке программ. Дополнительные драйверы обычно поставляются вместе с устройством или приобретаются у фирмы-производителя дополнительно.

## Контекст устройства

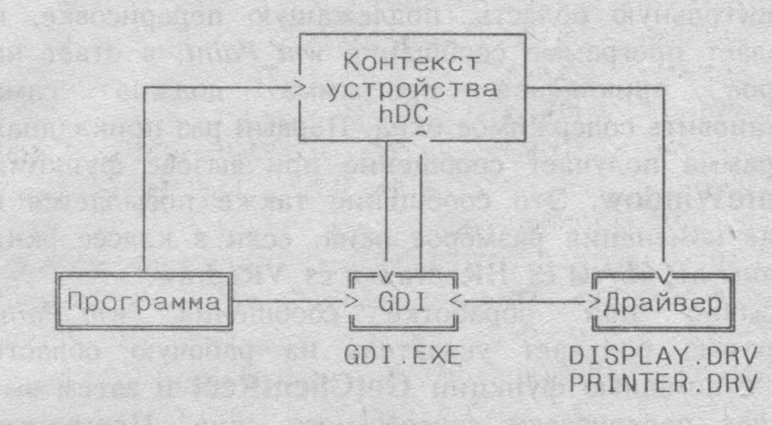
Функции GDI используют для обмена информацией с драйвером устройства специальную структуру данных — **контекст устройства**. Контекст содержит описание основных характеристик устройства и используемых средств отображения — кистей, карандашей, шрифтов и т.п.

Перед вызовом функции, выполняющей отображение информации, прикладная программа должна предоставить ей контекст устройства, на котором эта информация будет отображена. Формально же при вызове из команды любой функции GDI в качестве одного из параметров указывается ссылка на контекст устройства.

Схема взаимодействия прикладной программы с устройством показана на рисунке.

Поскольку контекст занимает довольно большой объем памяти, в Windows одновременно может быть доступно не более пяти контекстов, которые размещаются в специальном буфере, поддерживаемом ядром Windows. Таким образом, после того как программа отработала с контекстом (вывела порцию информации), она должна его освободить, чтобы он был доступен другим программам.

Чаще всего получение и удаление контекста выполняется при обработке сообщения *wm_Paint* вызовом пары функций **BeginPaint** и **EndPaint**.




Взаимодействие программы с устройством

```
MyDC := BeginPaint(MyWnd,MyPaint);
.....
  {Работа с контекстом}
.....
EndPaint(MyWnd,MyPaint);
```

Функция **BeginPaint** возвращает ссылку на контекст устройства, которая затем используется в качестве одного из параметров при вызове различных функций GDI.

> Function **BeginPaint** (Wnd : hWnd; var Paint : TPaintStruct) : hDC;
>
> Эта функция подготавливает окно к отображению информации в ответ на сообщение *wm_Paint*. Функция заполняет структуру типа TPaintStruct информацией, необходимой для работы функций отображения. Функция возвращает контекст экрана.

> Procedure **EndPaint**(Wnd : hWnd; var Paint : TPaintStruct);
>
> Эта процедура завершает операции по отображению в окне *Wnd*.

Переменная *MyPaint* — это структура типа *TPaintStruct*. Поле *hDC* этой структуры содержит ссылку на контекст, возвращенную функцией **BeginPaint**. Структура *TPaintStruct* также содержит структуру типа *TRect* (поле *rcPaint*), указывающую на прямоугольную область экрана, доступную прикладной программе для вывода информации. Доступ к этой области осуществляется с помощью ссылки на контекст. Вызов процедуры **EndPaint** завершает работу с определенной областью экрана и освобождает контекст.

Сообщение *wm_Paint* — одно из основных сообщений в жизненном цикле программы. С его помощью ядро Windows информирует программу о необходимости перерисовки части или всей рабочей области. Почему может возникнуть такая необходимость? Предположим, наше окно было частично перекрыто другим окном или панелью диалога. В результате часть рабочей области окна потеряла свое содержимое и должна быть обновлена. В этом случае ядро определяет прямоугольную область, подлежащую перерисовке, и посылает программе сообщение *wm_Paint*, в ответ на которое прикладная программа должна сама восстановить содержимое окна. Первый раз прикладная программа получает сообщение при вызове функции **UpdateWindow**. Это сообщение также посылается в случае изменения размеров окна, если в классе окна указаны атрибуты cs_HRedraw и cs_VRedraw.

Обычно при обработке сообщения *wm_Paint* программа получает указатель на рабочую область окна с помощью функции **GetClientRect** и затем выполняет перерисовку содержимого окна. Программа может сама стать источником сообщения *wm_Paint* — для этого достаточно вызвать функцию **InvalidateRect** или функцию **InvalidateRgn**.

Другим способом получения контекста является использование функции **GetDC**. Эта функция возвращает ссылку на контекст. В "паре" с ней используется функция **ReleaseDC**:

```
MyDC := GetDC(MyWnd);
.....
  {Работа с контекстом}
.....
ReleaseDC(MyWnd,MyDC);
```

В этом случае, в отличие от **BeginPaint** и парной к ней **EndPaint**, получаемый контекст определяет всю рабочую область экрана.

> Function **GetDC**(Wnd : hWnd) : hDC;
>
> Эта функция возвращает ссылку на контекст дисплея для указанного окна.

> Function **ReleaseDC**(Wnd : hWnd; DC : hDC) : Integer;
>
> Эта функция освобождает дисплейный контекст для указанного окна и делает его общедоступным.

Для получения контекста можно также использовать функцию **GetWindowDC**. Парной для нее является **ReleaseDC**:

```
MyDC := GetWindowDC(MyWnd);
.....
  {Работа с контекстом}
.....
ReleaseDC(MyWnd,MyDC);
```

Полученный контекст включает помимо рабочей области окна еще и область заголовка окна, меню, полосы прокрутки и окантовку.

> Function **GetWindowDC**(Wnd : hWnd) : hDC;
>
> Эта функция возвращает дисплейный контекст для всего окна (не только для его рабочей области).

Наконец, для получения контекста всего экрана может быть использована функция **CreateDC**. В качестве параметров указываются название драйвера (DISPLAY — для экрана), имя устройства, имя порта, к которому подключено устройство, и строка инициализации. В нашем случае (создание контекста для экрана) вызов этой функции будет выглядеть следующим образом:

```
MyDC := CreateDC('DISPLAY',0,0,0);
.....
  {Работа с контекстом}
.....
DeleteDC(MyDC);
```

Таблица 1

| Функция | | Описание |
|---|---|---|
| BeginPaint | EndPaint | Используется для перерисовки части рабочей области окна |
| GetDC | ReleaseDC | Используется для перерисовки всей рабочей области окна |
| GetWindowDC | ReleaseDC | Используется для перерисовки всего окна, включая области, занятые заголовком, рамкой и меню |
| CreateDC | DeleteDC | Используется для перерисовки всего экрана |

Таким образом, Windows предоставляет ряд различных функций для получения контекста. В табл. 1 кратко рассмотрены основные отличия описанных выше функций.

**Характеристики контекста**

Как уже отмечалось, контекст — это структура данных, описывающая некоторые характеристики устройства. Ниже приводятся список этих характеристик, их значения по умолчанию и функции, оказывающие воздействия на отдельные характеристики контекста (табл. 2).

Ниже мы рассмотрим ряд характеристик контекста более подробно.

Как и у художника, пишущего на холсте, в среде Windows имеется набор стандартных средств отображения графических изображений в текущем контексте — кисть, карандаш и шрифт.

Функция **GetStockObject** позволяет выбрать то стандартное средство, которое будет использоваться в данном контексте. Отметим, что в каждый момент времени может быть использован только один тип карандаша, кисти и шрифта, но после того как часть изображения выведена, средство можно поменять и продолжить отображение.

**Таблица 2**

| Атрибут | Значение по умолчанию | Функция GDI |
|---|---|---|
| Режим отображения | mm_Text | SetMapMode |
| Точка отсчета для окна | (0,0) | SetWindowOrg |
| Точка глобального отсчета | (0,0) | SetViewportOrg |
| Расширение окна | (1,1) | SetWindowExt |
| Глобальное расширение | (1,1) | SetViewportExt |
| Карандаш | Black_Pen | CreatePen<br>CreatePenIndirect<br>SelectObject |
| Шаблон фона | White_Brush | CreateBrushIndirect<br>CreateDIBPatternBrush<br>CreateHatchBrush<br>CreatePatternBrush<br>CreateSolidBrush<br>SelectObject |
| Шрифт | System_Font | CreateFont<br>CreateFontIndirect<br>SelectObject |
| Растровое изображение | | CreateBitmap<br>CreateBitmapIndirect<br>CreateComptibleBitmap<br>SelectObject |
| Текущее положение карандаша | (0,0) | MoveTo |
| Режим фона | Opaque | SetBkMode |
| Цвет фона | White | SetBkColor |
| Цвет текста | Black | SetTextColor |
| Режим рисования | R2_CopyPen | SetROP2 |
| Режим масштабирования | BlackOnWhite | SetStretchBltMode |
| Режим заливки многоугольников | Alternate | SetPolyFillMode |
| Межсимвольный размер | 0 | SetTextCharacterExtra |
| Отсчет для шаблона фона | (0,0) | SetBrushOrg<br>UnrealizeObject |
| Отсекающий весь экран прямоугольник | | ExcludeClipRect<br>IntersectClipRect<br>OffsetClipRgn<br>SelectClipRgn |
| Палитра | DEFAULT_PALETTE | CreatePalette<br>RealizePalette<br>SelectPalette |

В Windows стандартизованы средства отображения, перечисленные в табл. 3.

## Карандаш

Карандаш позволяет рисовать прямые линии, многогранники, окружности, дуги эллипсов и т.п. По умолчанию используется карандаш *Black_Pen*. Каждый карандаш позволяет проводить линии определенного цвета, ширины и стиля. Каждый из стандартных карандашей может иметь один из перечисленных в табл. 4 стилей.

Отметим, что карандаш *Null_Pen* служит для рисования невидимых линий.

Для использования карандаша в прикладной программе сначала необходимо получить ссылку на него (тип данных hPen). Например, для выбора карандаша *White_Pen* нужно вызвать функцию **GetStockObject**:

```
Pen := GetStockObject(White_Pen);
```

Получив ссылку на карандаш, мы выбираем его для использования в текущем контексте с помощью функции **SelectObject**:

```
SelectObject(MyDC,Pen);
```

**Таблица 3**

| Объект | Допустимое значение |
|---|---|
| Карандаш | *Black_Pen*<br>*Null_Pen*<br>*White_Pen* |
| Кисть | *Black_Brush*<br>*DkGray_Brush*<br>*Gray_Brush*<br>*Hollow_Brush*<br>*LtGray_Brush*<br>*Null_Brush*<br>*White_Brush* |
| Шрифт | *ANSI_Fixed_Font*<br>*ANSI_Var_Font*<br>*Device_Default_Font*<br>*OEM_Fixed_Font*<br>*System_Fixed_Font*<br>*System_Font* |

**Таблица 4**

| Стиль | Описание |
|---|---|
| *ps_Solid* | ________ |
| *ps_Dash* | _ _ _ _ |
| *ps_Dot* | . . . . |
| *ps_DashDot* | _._._. |
| *ps_DashDotDot* | _.._.. |
| *ps_Null* | Невидимый |
| *ps_InsideFrame* | Используется для рисования внутри рамки многогранников и ломаных линий |

Теперь карандашом *White_Pen* можно рисовать.

Отметим, что получение ссылки и выбор карандаша допустимо объединить:

```
SelectObject(MyDC,GetStockObject(White_Pen));
```

Функция возвращает ссылку на карандаш, который был текущим до ее вызова, и эта ссылка может быть использована для восстановления типа карандаша по окончании отображения:

```
Pen := SelectObject(MyDC,GetStockObject(White_Pen));
.....
{Отображение линий}
.....
SelectObject(MyDC,Pen);
```

Помимо стандартных возможно также создание собственных карандашей. Это полезно при рисовании линий различными стилями и цветами.

Для создания и использования карандаша нужно:

- вызвать функцию CreatePen, указав в качестве параметров стиль, ширину и цвет карандаша:

```
Pen := CreatePen(ps_Dot, 2, RGB(50,50,50))
```

- выбрать карандаш, включив его в текущий контекст:

```
SelectObject(MyDC,Pen)
```

- по окончании использования данного карандаша удалить его функцией DeleteObject:

```
DeleteObject(Pen)
```

> **Примечание:** Удаление всех создаваемых прикладной программой объектов является одним из условий ее *нормального функционирования*. Предопределенные в системе объекты не удаляются.

### Текущее положение карандаша

Текущее положение карандаша — это одна из характеристик контекста, определяющая то место в рабочей области окна, где будет выводиться следующая точка линии. Для определения текущей позиции карандаша используется функция **GetCurrentPosition**, а для установки нового положения — функция перемещения карандаша — **MoveTo** (без отображения) и **LineTo** (с отображением).

## Кисть

**Кисть** — другое средство отображения, предоставляемое ядром Windows. Кисть используется для заполнения областей внутри замкнутых фигур. В Windows существует ряд функций для отображения таких фигур (табл.*5*.).

**Таблица 5**

| Функция | Фигура |
|---|---|
| Rectangle | Прямоугольник |
| Ellipse | Эллипс |
| RoundRect | Прямоугольник с закругленными углами |
| Chord | Дуга |
| Pie | Сектор круга |
| Polygon | Многогранник |
| PolyPoligon | Сложный многогранник |

Контур такой фигуры отображается текущим карандашом, а затем область внутри шаблона заполняется текущей кистью. По умолчанию используется кисть *White_Brush*. В среде Windows существует шесть стандартных кистей: *Black_Brush, DkGray_Brush, Gray_Brush, Hollow_Brush, LtGray_Brush, Null_Brush* и *White_Brush*.

Кисть для заполнения области выбирается точно так же, как и карандаш — с помощью функции **GetStockObject**:

```
Brush := GetStockObject(Gray_Brush)
SelectObject(MyDC,Brush)
```

Кисть — это растровое изображение размером 8x8 пикселов (такое изображение называется *шаблоном заполнения фона*). А заливка области состоит в том, что шаблон отображается необходимое число раз по горизонтали и вертикали.

Для создания кистей ядро Windows предоставляет четыре функции: *CreateSolidBrush*, *CreateHatchBrush*, *CreatePatternBrush* и *CreateBrushIndirect*.

При сплошной заливке области кисть создается функцией **CreateSolidBrush**, а в качестве параметра указывается только цвет заливки:

```
Brush := CreateSolidBrush(RGB(50,50,50))
```

Также можно создать кисть на основе одной из стандартных масок. Для этого при вызове функции **CreateHatchBrush** указываются маска и цвет закраски:

```
Brush := CreateHatchBrush(hs_Cross, RGB(50,50,50))
```

Параметр, определяющий маску, может принимать одно из следующих значений: *hs_Horizontal*, *hs_Vertical*, *hs_FDiagonal*, *hs_BDiagonal*, *hs_Cross* и *hs_Diagcross*.

Просмотреть имеющиеся в системе кисти вы можете, например, с помощью приводимой ниже программы FILL.PAS.

```
{
.c.Программа FILL.PAS - показ стандартных кистей
}

uses WinCrt,WinTypes,WinProcs;
var
 MyWin : hWnd;
 MyDC  : hDC;
 Brush : Array[0..3] of hBrush;
 Pen   : hPen;

Begin
  CmdShow := sw_ShowMaximized;InitWinCRT;
  ReadKey;
  MyWin := FindWindow('TPWinCrt',PChar(0));
  MyDC := GetDC(MyWin);

{ Показать имеющиеся кисти }

  Pen := CreatePen(ps_Solid,1,RGB(0,0,0));
  Brush[0] := CreateHatchBrush(hs_Horizontal,RGB(0,0,0));
  Brush[1] := CreateHatchBrush(hs_Vertical,RGB(255,0,0));
  Brush[2] := CreateHatchBrush(hs_FDiagonal,RGB(0,255,0));
  Brush[3] := CreateHatchBrush(hs_BDiagonal,RGB(0,0,255));

  SelectObject(MyDC,Pen);
  SelectObject(MyDC,Brush[0]);
  Rectangle(MyDC,10,10,100,200);

  SelectObject(MyDC,Pen);
  SelectObject(MyDC,Brush[1]);
  Rectangle(MyDC,110,10,200,200);

  SelectObject(MyDC,Pen);
  SelectObject(MyDC,Brush[2]);
  Rectangle(MyDC,210,10,300,200);

  SelectObject(MyDC,Pen);
  SelectObject(MyDC,Brush[3]);
  Rectangle(MyDC,310,10,400,200);

{}
  ReadKey;
  ReleaseDC(MyWin,MyDC);
  DoneWinCrt;

End.
```

Чтобы создать кисть на основе растрового изображения, можно воспользоваться функцией **CreatePatternBrush**. В качестве параметра при вызове этой функции указывается ссылка на растровое изображение, используемое в качестве шаблона кисти.

```
Brush := CreatePatternBrush(Bitmap)
```

После того как кисть создана тем или иным способом, ее выбирают для использования в текущем контексте функцией **SelectObject**:

```
SelectObject(MyDC, Brush)
```

Использованные кисти удаляют функцией **DeleteObject**:

```
DeleteObject(Brush)
```

## Режим закраски фона, цвет фона и цвет текста

В Windows существует два режима закраски фона — *Transparent* и *Opaque*. В первом режиме выводимая на экран информация не изменяет фона (накладывается на него), во втором случае фон перерисовывается. Используя функцию **GetBkMode**, можно определить текущий режим фона, а с помощью функции **SetBkMode** — установить новый режим. По умолчанию используется режим *Opaque*.

Цвет фона и текста — это атрибуты, определяющие отображение информации на экране. Цвет фона устанавливается функцией **SetBkColor**, текущий цвет фона определяется с помощью функции **GetBkColor**.

Цвет текста устанавливается функцией **SetTextColor**, а его текущее значение определяется с помощью функции **GetTextColor**. Мы рассмотрим использование этих функций более подробно при описании методов вывода текста.

## Сохранение контекста

Как стало ясно из предшествующего изложения, контекст является общим ресурсом: после того как получившая его программа отработала графические функции, она должна освободить контекст, предоставив его в распоряжение другим программам. Иными словами, если программа несколько раз выполняет вывод на экран, она должна каждый раз получать и освобождать контекст. Может оказаться полезным и изменение ряда параметров контекста, на-

пример, при его получении. Поэтому при создании класса окна рекомендуется указывать стиль *cs_OwnDC*, означающий, что каждое окно данного класса будет иметь собственный контекст. В этом случае при обработке сообщения *wm_Create* выполняется изменение параметров контекста:

```
Case Message of
 ....
   wm_Create:
     MyDC := GetDC(MyWnd);
     ....
     {изменение атрибутов контекста}
     ....

     {отображение информации}
     ....
     ReleaşeDC(MyWnd,MyDC);
   End;
```

Стиль класса окна *cs_OwnDC* воздействует только на контекст, полученный с помощью функций GetDC и BeginPaint. Использование этого стиля увеличивает объем занимаемой окном памяти на 200 байт (контекст располагается в куче модуля GDI).

## Функции отображения графических объектов

В этом разделе кратко рассматриваются функции GDI, используемые для отображения линий, дуг и заполненных фигур.

### Отображение линий

При отображении линий используется карандаш, выбранный для работы в текущем дисплейном контексте. Чаще всего для перемещения карандаша без нанесения линии используют функцию *MoveTo*, а для изображения линии — функцию *LineTo*. Обе функции изменяют значение одного из атрибутов дисплейного контекста — текущего положения.

Функция *MoveTo* переносит текущую позицию в указанную точку. Функция *LineTo* отображает линию, соединяющую текущую позицию с указанной точкой. В приводимом ниже фрагменте программы текущая позиция устанавливается в точке с координатами (100,100) и из нее проводится линия до точки с координатами (200,200):

```
....
MoveTo(MyDC, 100, 100);
LineTo(MyDC, 200, 200);
....
```

Для отображения ломаной линии используется функция *PolyLine*. Построение ломаной линии можно осуществить и последовательными вызовами функции *LineTo*. Использование функции *PolyLine* позволяет более эффективно отобразить ломаную линию, при этом текущее положение карандаша не изменяется. В качестве параметров функции *PolyLine* указывается ссылка на текущий контекст, адрес массива, содержащего координаты всех точек излома линии, и размер этого массива:

```
procedure PolyDraw;
var
    PolyPoints : Array[0..3] of TPoint;
begin
    PolyPoints[0].X := 100;   PolyPoints[0].Y := 100;
    PolyPoints[1].X := 150;   PolyPoints[1].Y := 150;
    PolyPoints[2].X := 200;   PolyPoints[2].Y := 200;
    PolyPoints[3].X := 250;   PolyPoints[3].Y := 250;
    PolyLine(MyDC,@PolyPoints,4);
end;
```

**Примечание:** Количество точек для построения ломаной линии должно быть не менее двух.

### Отображение дуг

Для отображения дуг используется функция *Arc*. Эта функция строит дугу эллипса. Центром эллипса считается центр охватывающего прямоугольника, левый верхний угол которого имеет координаты (wX1, wY1), а правый нижний — координаты (wX2, wY2). Дуга отображается текущим карандашом из точки с координатами (wX3, wY3) в точку с координатами (wX4, wY4) против часовой стрелки. Если начальная и конечная точки имеют одинаковые координаты, вместо дуги отображается полный эллипс (окружность — в случае, если охватывающий прямоугольник является квадратом).

В табл. 6 кратко описаны функции GDI для отображения линий и дуг.

## Функции отображения заполненных фигур

Рассматриваемые ниже функции используют определенные в контексте карандаш (для отображения окантовки фигуры) и кисть (для ее заполнения). Текущая позиция карандаша при этом не изменяется.

### Прямоугольник

Прямоугольники отображаются функцией *Rectangle*. При этом в качестве параметров указываются координаты его левого верхнего и правого нижнего углов:

```
Rectangle(MyDC,100,100,200,200);
```

Для построения прямоугольников с закругленными углами используется функция *RoundRect*. Углы у таких прямоугольников отображаются как 1/4 эллипса. В качестве параметров при вызове этой функции указываются координаты левой верхней точки и правой

**Таблица 6**

| Функция | Описание |
|---|---|
| Arc | Отображает дугу |
| LineDDA | Вычисляет точки для построения линии |
| LineTo | Отображает прямую линию |
| MoveTo | Перемещает карандаш в указанную точку без рисования |
| Polyline | Строит ломаную линию |

нижней точки прямоугольника и ширина и высота эллипса, используемого для отображения закругленных углов.

### Эллипс

Для построения эллипсов используется функция *Ellipse*. В качестве параметров этой функции указываются координаты охватывающего прямоугольника.

Для построения фрагментов эллипсов используются функции *Pie* и *Chord*. Обе эти функции отображают дуги, но, в отличие от функции *Arc*, после построения окантовки дуги ее внутренняя часть заполняется текущей кистью. Функция *Pie* соединяет центр эллипса с конечными точками дуги, функция *Chord* соединяет две точки, принадлежащие дуге, прямой линией.

### Многогранник

Для отображения заполненных многогранников используется функция *Polygon*. С помощью этой функции отображаются связанные между собой прямые линии (как и в случае использования функции PolyLine), но в данном случае конечные линии соединяются, образуя замкнутую фигуру.

В табл. 7 кратко описаны функции GDI, используемые для отображения заполненных фигур.

**Таблица 7**

| Функция | Описание |
|---|---|
| Chord | Отображает дугу эллипса |
| DrawFocusRect | Отображает прямоугольник |
| Ellipse | Отображает эллипс |
| Pie | Отображает сектор круга |
| Polygon | Отображает многогранник |
| PolyPoligon | Отображает сложный многогранник |
| Rectangle | Отображает прямоугольник |
| RoundRect | Отображает прямоугольник с закругленными углами |

В качестве примера приведем программу FIGURES.PAS, показывающую, как использовать рассмотренные функции отображения различных геометрических фигур.

```
{
.c. FIGURES.PAS Программа для отображения различных
геометрических фигур
}

uses WinCrt,WinTypes,WinProcs;
var
 MyWin : hWnd;
 MyDC  : hDC;
 Brush : hBrush;
 Pen   : hPen;

Begin
  CmdShow := sw_ShowMaximized;InitWinCRT;
  ReadKey;
  MyWin := FindWindow('TPWinCrt',PChar(0));
  MyDC := GetDC(MyWin);

{-- Выбрать инструменты для рисования --}

  Pen := GetStockObject(Black_Pen);
  Brush := GetStockObject(LtGray_Brush);
  SelectObject(MyDC,Brush);
  SelectObject(MyDC,Pen);

{-- Нарисовать различные фигуры --}

  Rectangle(MyDC,50,50,100,100);
  RoundRect(MyDC,50,150,100,200,15,15);
  Ellipse(MyDC,150,50,200,100);
  Chord(MyDC,150,150,200,200,150,150,200,200);
  Pie(MyDC,250,50,300,100,250,50,300,50);

  ReadKey;
  ReleaseDC(MyWin,MyDC);
  DoneWinCrt;
End.
```

*Д.Рогаткин, А.Федоров*



# Организация защиты вычислительных систем

## 4. Реализация политики безопасности

После того как определены основные цели защиты и пути их достижения, а значит, определена политика безопасности, необходимо ее реализовать на выбранной в результате анализа риска базе. Реализация политики означает настройку механизмов реальных систем на конкретные требования. Любая система обычно обладает достаточно гибкими механизмами, обеспечивающими широкий выбор возможностей. Поэтому реализацию должны проводить специалисты, во-первых, хорошо знакомые с принципами планирования защиты (см. предыдущие этапы), а во-вторых, досконально знающие выбранную систему и другие средства, особенно их возможности в области защиты. В противном случае существует риск образования брешей защиты — недостатков планирования или реализации, способных нарушить политику безопасности и стать своего рода воротами для проникновения и других нарушений. Подавляющее большинство аварий происходит не из-за "прорывов" системы, а из-за умелого использования нарушителями ее свойств, которые не были известны или учтены теми, кто планировал и реализовывал защиту. Во многих случаях в группу, занимающуюся такой работой, могут входить специалисты тех же профилей, что и при проведении анализа риска.

Главными при реализации политики безопасности являются определение защищаемых объектов, круга их пользователей и правил передачи полномочий на доступ к ценной информации.

Более подробно говорить о последовательности действий при реализации политики трудно, так как этот этап в значительной мере зависит от особенностей конкретной системы. Поэтому поговорим о свойствах механизмов защиты, характерных для большинства защищенных систем.

---

Окончание. Начало в КомпьютерПресс №10´92.

### 4.1. Механизмы реализации политики безопасности

Естественно предположить, что все системные средства, отвечающие за реализацию политики безопасности, в первую очередь сами должны быть защищены от любого вмешательства в их работу. В противном случае говорить о надежности защиты будет достаточно трудно. Можно изменять их параметры, но они сами должны оставаться в неприкосновенности. Поэтому все средства защиты и контроля реализованы в *достоверной (или надежной) вычислительной базе* (НВБ). Это полностью защищенный механизм вычислительной системы, включая аппаратные и программные средства, комбинация которых отвечает за соблюдение политики безопасности. НВБ состоит из компонентов, совместная работа которых проводит единую политику над программным продуктом или системой. Способность НВБ правильно работать зависит от ее механизма и корректного управления.

Основой НВБ является ядро безопасности. Это элементы техники и программного обеспечения, разделяющие все попытки доступа субъектов к объектам, защищенные от модификации и проверенные на корректность.

Именно тот факт, что ядро безопасности перехватывает и проверяет любое обращение к объектам, является гарантией надежности (если само ядро надежно). Дело в том, что программы ядра функционируют на самом низком уровне как самые привилегированные в системе и ни один запрос не может миновать их. Этим ядро безопасности отличается от защиты с помощью прикладных средств, где наверняка существуют возможности преодоления защиты путем выхода на более низкий уровень или в привилегированный режим. А ниже уровня ядра безопасности опускаться уже некуда.

Ядро безопасности в разных системах может функционировать по-разному — перехватывать любое обращение к объекту, производить выборочную про-

верку прав доступа, проверять обращения к определенным объектам и т.д. Все это должно быть хорошо известно специалистам по защите.

Другим важным механизмом реализации политики безопасности является *концепция защищенной области*. Она заключается в объединении объектов по их местонахождению, хранимой информации, защите и внутренней структуре. Управление (в том числе доступом) осуществляется областью в целом с помощью специального программного обеспечения, доступ к отдельным объектам запрещен. Области могут быть разных уровней, они могут вкладываться друг в друга, соприкасаться друг с другом, быть независимыми, но не пересекаться. Примерами защищенных областей может служить защищенная база данных со своими наборами и программным обеспечением; компьютер, содержащий базу данных, со своей ОС; защищенная сеть, содержащая компьютер.

Некоторые ОС сами реализуют концепцию защищенной области, в других это задача планировщика защиты.

## 4.2. Принципы реализации политики безопасности

Как уже отмечалось выше — настройка механизмов защиты — дело сугубо индивидуальное для каждой системы и даже для каждой задачи. Поэтому рассказать о том, как это делается, довольно трудно. Однако существуют общие принципы, проверенные практикой, которых надо придерживаться, чтобы облегчить себе работу. Рассмотрим их.

**Группирование** — это представление множества субъектов одним групповым именем; всем субъектам, принадлежащим одной группе, предоставляются равные права. Принципы объединения пользователей в группы могут быть самые разные: ссылки на одни и те же объекты, одинаковый характер вычислений, работа над совместным проектом и т.д. При этом один и тот же субъект может входить в несколько групп и соответственно иметь различные права по отношению к одному и тому же объекту.

Механизм группирования может быть иерархическим. Это означает, что каждый субъект является членом нескольких групп, упорядоченных по отношению “быть подмножеством“.

Контроль за состоянием групп очень важен, поскольку члены одной группы имеют доступ к большому числу объектов, что не способствует их безопасности. Создание групп и присвоение групповых привилегий должно осуществляться администратором безопасности, руководителем группы или каким-либо другим лицом, несущим ответственность за сохранность групповых объектов.

**Правила умолчания.** Большое внимание при назначении привилегий следует уделять правилам умолчания, принятым в данной системе, — это необходимо для соблюдения политики безопасности. Во многих системах, например, субъект, создавший объект и являющийся его владельцем, по умолчанию получает все права на него, кроме того, он может эти права комулибо передавать.

В каждой ОС используются свои правила умолчания, однако принципы назначения привилегий по умолчанию в большинстве систем одни и те же. Если в системе используется древовидная файловая структура, то необходимо принимать во внимание правила умолчания для директорий. Корректное использование правил умолчания способствует поддержанию целостности политики безопасности.

**Минимум привилегий** — один из основополагающих принципов любой политики безопасности, используемый повсеместно. Каждый пользователь и процесс должны иметь *минимальное* множество привилегий, необходимых для работы. Определение для всех пользователей множества привилегий, с одной стороны, позволяющих осуществлять быстрый доступ ко всем необходимым для работы объектам, а с другой стороны — запрещающих доступ к чужим объектам — проблема достаточно сложная. От ее решения во многом зависит корректность реализации политики безопасности.

**“Надо знать“.** Этот принцип во многом схож с предыдущим. В соответствии с ним полномочия пользователей назначаются согласно их обязанностям — доступ разрешен только к той информации, которая необходима им для работы.

**Централизация критичной информации.** Многие системы практикуют сбор, хранение и обработку информации одного уровня в одном месте (узле сети, устройстве, директории). Это связано с тем, что проще защитить одним и тем же способом большой массив информации, чем организовывать индивидуальную защиту для каждого набора. Когда информация собрана в одном месте, к ней легче контролировать доступ; при этом также снижается вероятность ошибки в защите.

Для реализации этого принципа могут быть разработаны специальные программы, управляющие обработкой таких наборов. Это будет простейший способ построения защищенных областей.

**Иерархия привилегий.** Контроль объектов системы может иметь иерархическую организацию. Такая организация принята в большинстве коммерческих систем. При этом схема контроля имеет вид дерева, в котором узлы — субъекты системы, ребра — права на контроль привилегий согласно иерархии. Корнем является администратор системы, имеющий право изменять привилегии любого пользователя. Узлами низших уровней являются администраторы подсистем, имеющие права изменять привилегии пользователей этих подсистем (в их роли могут выступать руководители организаций, отделов, если подсистема обеспечивает работу этой организации, отдела и т.д.), а листьями дерева являются все пользователи системы. Вообще говоря, субъект, стоящий в корне любого поддерева, имеет право изменять защиту любого субъекта, принадлежащего этому поддереву.

Достоинство такой структуры состоит в том, что она точно копирует схему организации, которую обслуживает система, вследствие чего легко составить множество субъектов, имеющих право контролировать данный объект. Недостаток же ее в том, что множество субъектов может быть довольно большим.

**Привилегии владельца.** При таком контроле каждому объекту соответствует единственный субъект с исключительным правом контроля объекта — владелец (обычно это тот, кто его создал). Владелец обладает всеми разрешенными для этого типа данных правами на объект, он может разрешать доступ к объекту любому другому субъекту, но не имеет права никому передать привилегию на корректировку защиты. Однако такое ограничение не касается администраторов системы — они имеют право изменять защиту любых объектов.

Главный недостаток принципа привилегии владельца состоит в том, что при обращении к объекту пользователь должен предварительно получить разрешение у владельца (или администратора), что может затруднять работу (например, при отсутствии владельца или просто его нежелании разрешить доступ). Поэтому такой принцип обычно используется при защите личных объектов пользователей.

**Свободная передача привилегий.** При такой схеме субъект, создавший объект, может передать любые права на него любому другому субъекту вместе с правом корректировки средств контроля доступа к этому объекту, а тот, в свою очередь, может передать все эти права дальше — любому субъекту.

Естественно, при этом возникают большие трудности в определении круга субъектов, имеющих в данный момент доступ к объекту (права на объект могут распространяться очень быстро и так же быстро исчезать), и поэтому такой объект легко подвергнуть несанкционированной обработке. В силу этих обстоятельств такая схема применяется достаточно редко — в основном, в исследовательских группах, работающих над одним проектом, — когда все имеющие доступ к объекту заинтересованы в его содержимом.



Перечисленные принципы в чистом виде применяются в вычислительных системах редко; обычно используются разнообразные их комбинации. Ограничение доступа к объектам в ОС включает в себя ограничение доступа к некоторым системным возможностям, например, некоторым командам, программам и т.д., если при их использовании нарушается политика безопасности. Вообще, набор полномочий каждого пользователя должен быть тщательно продуман, из него должны быть исключены возможные противоречия и дублирования, поскольку именно здесь кроется основная причина большинства нарушений. Чаще всего утечка информации происходит в системах с плохо спроектированной или реализованной политикой безопасности, в обход существующей защиты.

### 4.3. Защита сети

При защите сети надо руководствоваться теми же принципами, что и при защите ОС. Однако здесь есть свои особенности.

Главная особенность состоит в том, что для надежной защиты сети недостаточно ограничиться защитой каждого из ее узлов или групп узлов. Ведь защищенная сеть — это единый комплекс, а защита отдельных узлов отвечает лишь за его часть. Основная сложность в том, чтобы связать два или несколько узлов в сеть с гарантией надежности. Так, при запросе доступа по сети ядру безопасности одного узла придется иметь дело с фиктивным субъектом (он находится на другом узле, под управлением ядра безопасности другой ОС), а ядру безопасности другого узла — с фиктивным объектом. С другой стороны, эти две системы должны иметь единое ядро безопасности.

Дополнительные сложности возникают в ситуации, когда два узла имеют разные уровни безопасности (согласно полномочной политике), поскольку здесь приходится еще следить за соответствием уровней.

Те же проблемы возникают и с другими механизмами защиты при организации сети.

Большинство систем имеют в своем распоряжении средства, которые помогают успешно решить все упомянутые проблемы, не создавая затруднений пользователям и не снижая надежности. Надо только позаботиться о том, чтобы защита сети планировалась как защита единого комплекса вычислительных средств. Здесь уже нельзя обойтись элементарными средствами вроде доступа к узлу по паролю — сколько один пользователь может запомнить паролей?!

## 5. Контроль защиты

Контроль за состоянием системы — один из ключевых моментов поддержания нормальной работы защиты. Даже лучшая система, спроектированная и реа-

лизованная без ошибок, может преподнести немало неприятных сюрпризов, если за ней постоянно не следить.

### 5.1. Источники нарушений

Источники нарушения защиты вычислительных систем можно разбить (достаточно условно) на три категории:

1. Безответственность пользователя — относится к ситуациям, в которых пользователь целенаправленно или случайно производит какие-либо действия, связанные с нарушением политики безопасности, но не ставящие под угрозу работоспособность системы. Например, копирование содержимого определенного файла с целью продажи.

2. Зондирование системы пользователем — это те ситуации, когда пользователь умышленно эксплуатирует недостаточно защищенные части системы. Некоторые пользователи считают получение доступа к системным наборам крупным успехом, затевая своего рода игру "пользователь-против-системы". Хотя намерения могут быть и безвредными, эксплуатация ресурсов считается нарушением. Пользователи с более определенными намерениями могут найти конфиденциальные данные, попытаться испортить их или даже уничтожить. И здесь уже все гораздо серьезнее.

3. Проникновение пользователя — это ситуации, когда пользователь преодолевает защиту для получения доступа к системе. Несмотря на то что многие системы имеют средства, делающие такое проникновение чрезвычайно сложным, создать полностью защищенную ОС практически невозможно.

Тот, кому успешно удалось проникновение — одновременно очень квалифицирован и опасен, так как проникновение — это наиболее серьезный и опасный вид нарушений.

### 5.2. Два вида контроля

Надо знать, что от всех напастей не застрахуешься, поэтому лучше всего выделить основные, представляющие наибольшую опасность для вас, и защищаться именно от них. Это делается на этапе анализа риска.

Анализ риска определяет возможные виды угроз системе и оптимальные меры защиты. Эти меры можно разделить на два основных типа:
- системные;
- административные.

Системные меры защиты обусловлены концепцией "надежного программного обеспечения", которое реализовано с помощью НВБ. Эти меры гарантируют, что те требования, которые заложены в систему, будут выполняться безошибочно.

Понятно, что могут представлять из себя административные меры — пропускной режим, замок в компьютерном зале, сейфы для лент и дисков и т.д. Эти меры также определяются в процессе анализа риска. Но кроме того, к административным мерам относятся также действия операторов и администраторов по предотвращению попыток нарушения работы системы. Их мы рассмотрим подробнее.

### 5.3. Опасные события

Чтобы предотвратить или исправить какую-либо неприятность, случившуюся с вашей системой, надо прежде всего хорошо представлять себе, что с ней вообще может случиться. Для большинства систем круг событий, которые мы будем называть *опасными событиями*, один и тот же. К ним можно отнести:
- попытки захвата пароля;
- попытки проникновения в систему;
- изменение базы данных защиты;
- получение привилегий;
- вирусные атаки (включая внедрение "троянских коней", "червей" и т.д.);
- угрожающее использование ресурсов (памяти, времени процессора и др.);
- проникновение через сеть.

В случае наступления любого из этих опасных событий должны предприниматься немедленные меры для предотвращения крушения системы и сохранения данных.

### 5.4. Появление угрозы

Информация о том, что систему пытаются атаковать, может быть получена из следующих источников.

1. Сообщения пользователей. Если кто-то из них заметил, что, например, дата последнего входа в систему не соответствует действительности, или пропал какой-то файл, или произошла любая другая неприятность, он должен первым делом обратиться к оператору или администратору.



2. Мониторинг системы. Оператор в процессе работы должен периодически контролировать систему с помощью соответствующих средств — следить за загруженностью процессора, состоянием очередей, использованием памяти. Практически любая угроза отражается на состоянии системных ресурсов.

3. Ведение системного журнала. Он должен отслеживать такие события, как доступ к наиболее охраняемым объектам, входы и выходы из системы (особенно с использованием сети), попытки проникновения, обращения к периферийным устройствам и т.д. Системный журнал собирает сведения, которые могут представлять интерес либо с точки зрения статистики, либо с точки зрения непосредственного отношения к угрозе безопасности системы.

Разумное сочетание этих мер поможет избежать многих опасностей — уж где-нибудь злоумышленник да проявит себя, важно лишь его не пропустить.

В некоторых случаях обычных мер контроля, предлагаемых системой, может оказаться недостаточно. Тогда применяют специально разработанные дополнительные меры. К ним можно отнести, во-первых, статистические меры. Специальные программы постоянно следят за состоянием некоторых параметров системы, анализируя их. При появлении каких-либо отклонений сразу выдается тревожное сообщение. Это своего рода автоматизация мониторинга системы.

Во-вторых, дополнительными мерами контроля могут служить некоторые интеллектуальные средства. Если система имеет достаточно большие размеры, следить за состоянием ее защиты трудно. Поэтому можно установить специальные программные средства, которые будут настроены на анализ определенных состояний системы, например, на опасные события или изменение конфигурации. Они могут вовремя сообщить о появлении возможности НСД или каналов утечки информации, которые оперативно обнаружить обычным способом весьма непросто.

## 6. Заключение

Из всего сказанного можно сделать вывод, что организация защиты компьютерной системы — это единый комплекс мер, которые должны учитывать все аспекты процесса обработки информации. Несмотря на некоторые неудобства, причиняемые пользователю во время работы, во многих случаях средства защиты могут оказаться совершенно необходимыми для нормальной жизнедеятельности системы.

Перечислим основные неудобства защиты:
- с большинством защищенных систем достаточно трудно работать;

- увеличение защищенности системы может привести к увеличению накладных расходов при обработке информации в связи с замедлением доступа к данным, замедлением выполнения операций, замедлением системной работы;
- увеличение защищенности требует увеличения рабочего времени пользователя для выполнения одного и того же задания.

Что же касается необходимости средств защиты, принцип "пока гром не грянет, мужик не перекрестится" в этом вопросе совершенно себя не оправдывает — порой на карту может быть поставлено слишком многое. По крайней мере, совершенно необходимо провести хотя бы минимальные исследования (анализ риска) и установить "защиту от дурака". Те же, кому необходимы средства защиты с гарантированными свойствами, должны провести полный анализ системы с учетом "Перечня апробированной продукции" (EPL) и "Перечня рекомендуемой продукции" (PPL) [1], которые составлены NCSC [2] специально для этих целей. Свойства программного обеспечения определяются стандартом "Trusted Computer System Evaluation Criteria" ("Оранжевая книга") [3].

В заключение отметим, что при реализации цепочки "требования — политика — модель — механизмы — реализация — поддержка и контроль" необходимо учитывать следующее.

1. Прежде всего, определите цели защиты — нужна ли она вообще? Проведите анализ риска — он поможет подобрать оптимальные средства защиты, и составьте план. В конечном счете это все равно окажется дешевле, чем действовать наобум.

2. Критически отнеситесь к требованиям защиты — не чрезмерны ли они? Систем, защищенных "на сто процентов", не бывает. Можно лишь минимизировать риск.

3. Любая система реализует лишь те требования, которые заложены в ее модель. Знание модели поможет понять, что можно ожидать от вашей системы.

4. При реализации защиты нужны специалисты, знакомые как с теорией, так и с практикой организации защиты для каждой конкретной системы. В противном случае существует риск образования брешей защиты.

5. В подавляющем большинстве нарушений, вплоть до крушений, виновата не система, а ошибки планирования и реализации.

6. Предусмотреть все опасности никогда не удастся. Лучше предусмотреть самые серьезные и обезопасить от них самую ценную информацию.

7. В работе необходимо уделять должное внимание контролю — почти всякая угроза обязательно обнаружит себя. Вовремя обнаруженная опасность уменьшит потери.

*А.Першин*

[1] EPL (Evaluated Products List) — список продукции, содержащий операционные системы, подсистемы защиты и дополнительные средства защиты.

PPL (Preferred Products List) — список оборудования, включающий весь спектр техники (компьютеры, рабочие станции, периферийное оборудование и т.д.).

Данная продукция рекомендована, а в некоторых случаях обязательна к использованию в защищенной среде.

[2] NCSC (National Computer Security Center) — Национальный Центр Компьютерной Безопасности — государственная организация США, учрежденная в 1981 году для координации усилий по защите информации и предотвращению компьютерных преступлений.

[3] Стандарт, содержащий требования к средствам обеспечения безопасности компьютерных систем. Согласно этим требованиям все системы делятся на группы и классы безопасности в зависимости от их свойств. Подтверждение выполнения тех или иных требований осуществляется NCSC; в случае положительного исхода проверки данной системе присваивается соответствующий рейтинг безопасности.

**Использованы материалы:**


1. "Зарубежная радиоэлектроника", №12, 1989.
2. S. Muftic, "Security Mechanisms for Computer Networks", Ellis Horwood Limited, 1989.
3. Datapro Report on Information Security, 1989, 1991.
4. D. P. Downs, J. R. Rub and others, "Issues in Discretionary Access Control", IEEE Proc. of the 1985 Symp. on Security and Privacy, pp. 208-218.
5. J. K. Millen, "Covert Channel Capacity", IEEE Proc. of the 1987 Symp. on Security and Privacy, pp. 60-66.
6. J. Guttman, "Information Flow and Invariance", IEEE Proc. of the 1987 Symp. on Security and Privacy, pp. 67-73.
7. J. McLean, "Reasoning about Security Models", IEEE Proc. of the 1987 Symp. on Security and Privacy, pp. 123-131.

# Каталог продуктов фирмы NOVELL

## ПРОДУКТЫ СВЯЗИ С СЕТЯМИ АРХИТЕКТУРЫ IBM SNA

### NetWare SNA Gateway ELS

NetWare SNA Gateway ELS — это программный продукт, обеспечивающий наиболее простой метод создания шлюза и превращающий IBM PC, PS/2 и любую другую совместимую ПЭВМ в сервер-шлюз SNA. Продукт позволяет до 16 PC, PS/2 и рекомендованным совместимым ПЭВМ, входящим в ЛВС, осуществлять связь с большой ЭВМ архитектуры SNA фирмы IBM или совместимой. NetWare SNA Gateway ELS работает в сетях NetWare SPX или IBM NetBIOS и поддерживает следующие две опции связи с главной системой:

- коаксиальная. Опция поддерживает до пяти сеансов диалога и/или печати на принтере 3287 через коаксиальное соединение между адаптером эмулятора на сервере-шлюзе и портом DFT на кластерном контроллере 3x74;
- удаленная. Эта простая опция удаленной связи поддерживает до 16 сеансов диалога и/или печати, обеспечивая передачу данных от синхронного модема к фронтальному процессору 37xx со скоростью до 19,2 Кбит/с.

#### Характеристики

- Включает NetWare LU6.2 для поддержки прикладных программ LU6.2.
- Поддерживает до 16 сеансов связи с главной ЭВМ, доступных 16 пользователям сети.
- Утилита Gateway Status позволяет получать текущую информацию о состоянии сеансов с главной системой, линий связи, рабочих станций и идентифицировать сеансы (LU).
- Инструменты управления включают средство внутреннего отслеживания, позволяющее быстро проводить диагностику и внутренние тесты для локализации проблем, возникающих в ЛВС на участке между шлюзом и рабочей станцией.
- В результате всеобъемлющего опробования и тестирования в критических режимах работы, NetWare SNA Gateway ELS обеспечивает самый высокий уровень надежности среди продуктов в этой области промышленности.
- Использует протокол Service Advertising Protocol (SAP) фирмы Novell, обеспечивающий простой доступ к шлюзу.
- Конфигурация продукта может быть задана так, чтобы поддерживать связь с несколькими главными системами, что позволяет легко изменять соединения с главными системами по мере роста сети. Это также предоставляет возможности восстановления данных в результате стихийных бедствий.
- Обеспечивает возможность передачи файлов.
- Простота инсталляции обеспечивается полной и общедоступной документацией, изложенной в руководстве Quick-Start Guide. Включает программу адаптации, использующую интерфейс меню.

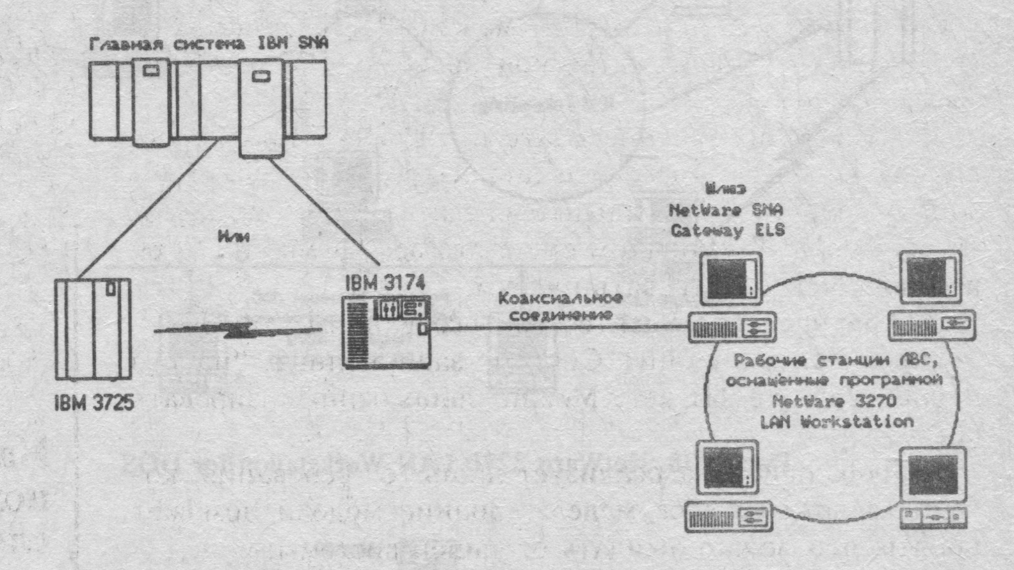



Рис. 3.31. NetWare SNA Gateway ELS: коаксиальное соединение

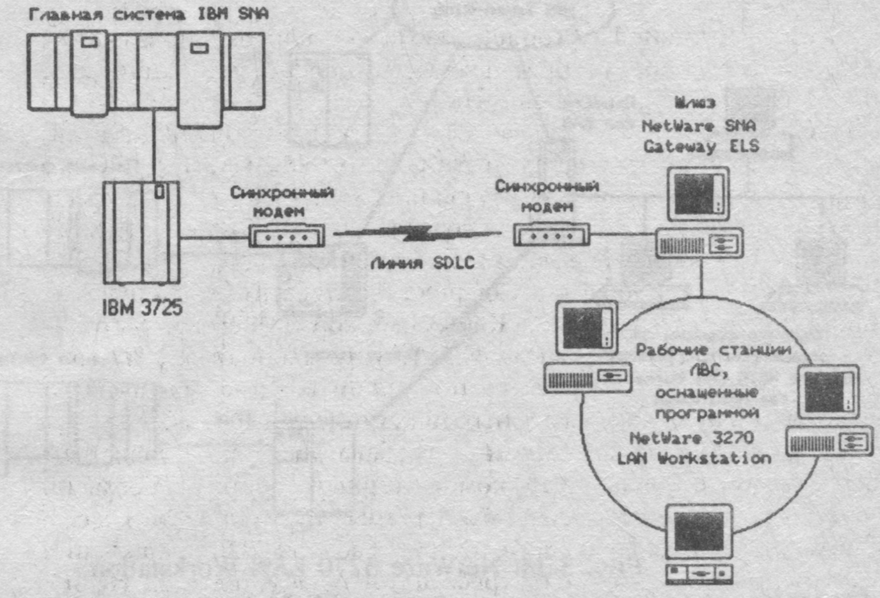



Рис. 3.32. NetWare SNA Gateway ELS: удаленное соединение

Материал подготовлен совместно СП "ИнтерПроКом" (Тел.: (095) 129-80-09, 129-80-33) и агентством КомпьютерПресс. Начало в №№3-10'92.

- В одной ЛВС может работать несколько шлюзов, что позволяет рабочим станциям осуществлять доступ к нескольким главным ЭВМ. NetWare SNA Gateway позволяет также реализовать резервные шлюзы, которые обеспечивают резервирование и разделение нагрузки между несколькими шлюзами, подключенными к интенсивно используемой главной системе.
- Поддерживает пулы LU и выделенные LU (логические устройства). Доступ к пулам LU осуществляется пользователями по принципу "первый пришел — первый обслуживается". Выделенные LU позволяют супервизору сети присваивать LU конкретным пользователям.
- Оптимизирован для использования в сетях NetWare IPX.
- Шлюз может быть как выделенным, так и использоваться одновременно в качестве рабочей станции (если позволяет память).

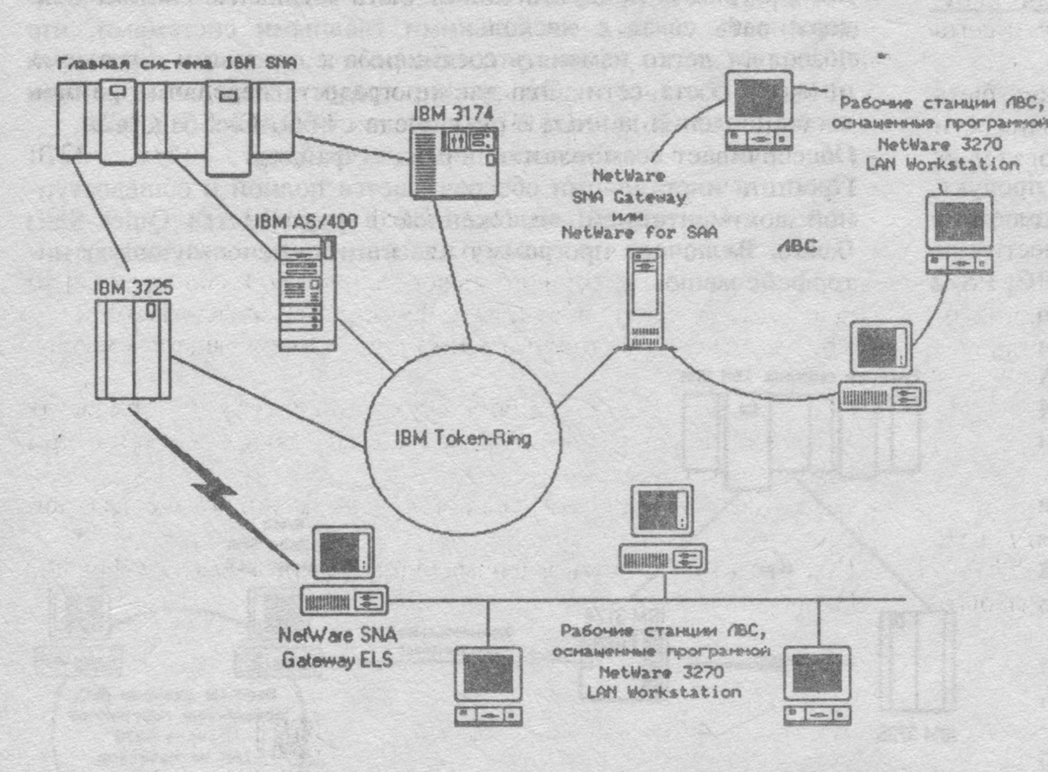



Рис. 3.33. NetWare 3270 LAN Workstation for DOS

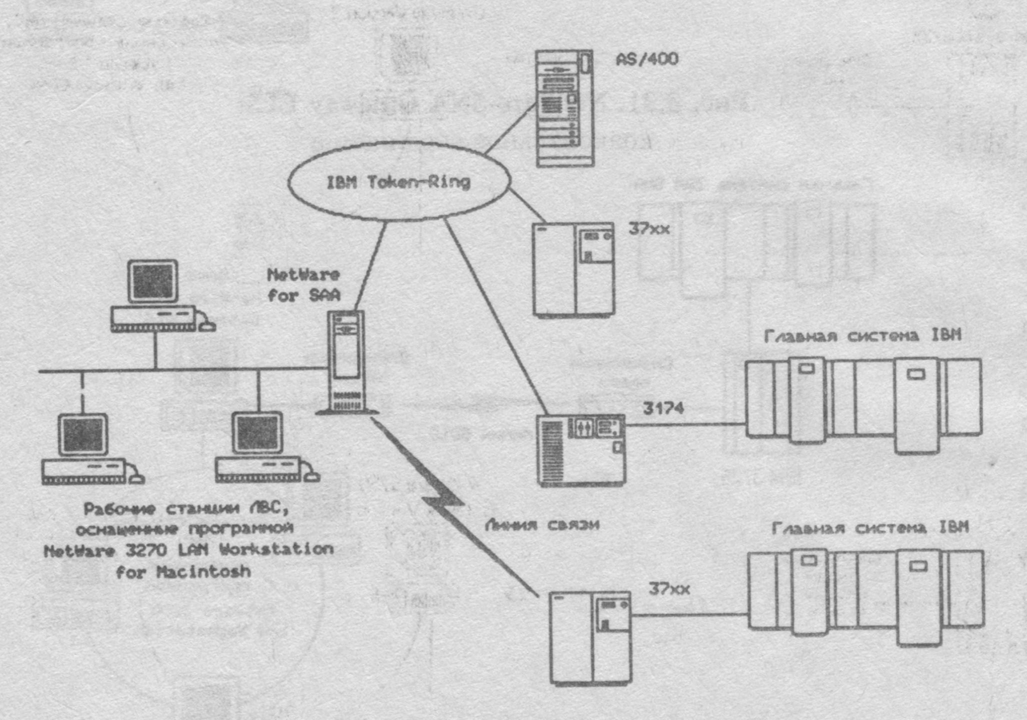



Рис. 3.34. NetWare 3270 LAN Workstation for Macintosh позволяет пользователям ЛВС иметь доступ к главной системе IBM через NetWare for SAA

## NetWare 3270 LAN Workstation for DOS

NetWare 3270 LAN Workstation for DOS — пакет программ эмуляции терминалов и принтеров IBM, используемый совместно с NetWare SNA Gateway, NetWare SNA Gateway ELS или NetWare for SAA. NetWare for SAA требует использования NetWare 3270 LAN Workstation for DOS v2.0. Этот продукт обеспечивает PC, PS/2 и рекомендованным совместимым рабочим станциям недорогой доступ к большим ЭВМ и ЭВМ среднего размера через программы NetWare SNA Gateway или NetWare for SAA.

Программа эмуляции поддерживает до пяти сеансов с главными ЭВМ на одной рабочей станции, среди которых могут быть сеансы диалога, печати или сеансы усовершенствованной межпрограммной связи (Advanced Program-to-Program Communications — APPC), а также сеансы DOS, передача файлов или утилиты переназначения клавиатуры. Предназначенный для работы в сетях NetWare SPX или IBM NetBIOS (только для NetWare SNA Gateway), пакет NetWare 3270 LAN Workstation for DOS продается по лицензии на один сервер, что позволяет супервизору сети покупать только одну копию продукта для одного файлового сервера независимо от того, сколько рабочих станций он обслуживает.

NetWare 3270 LAN Workstation for DOS позволяет осуществлять передачу файлов на главную ЭВМ и с нее, используя утилиты SEND/RECEIVE или TRANSFER, которые входят в комплект поставки. Утилиты SEND/RECEIVE выгружают и загружают файлы из главной системы и в нее, осуществляя взаимодействие с программой передачи файлов главной ЭВМ IBM IND$FILE в средах TSO, CMS и CICS. Программа TRANSFER позволяет осуществлять обмен данными RFT и FFT между PC и главной системой при использовании прикладных программ SAA для автоматизации делопроизводства, таких как OfficeVision или DISOSS.

### Характеристики

- Включает NetWare LU6.2 для поддержки прикладных программ LU6.2.
- Поддерживает полноэкранные модели Model 2, 3, 4 и 5.
- Средство автоматического переподключения позволяет заново устанавливать соединение между рабочей станцией и шлюзом в случае отказа в работе последнего.
- Средство адаптации сеанса печати позволяет назначать в качестве принтера сетевой или локальный принтер, указывать количество строк на странице и настраивать программу так, что при вызове она автоматически инициирует сеанс печати с главной ЭВМ.
- Эмулирует терминалы 3270 и принтеры 3287 фирмы IBM.
- Поддерживает до пяти сеансов диалога и печати на одной рабочей станции ЛВС. При использовании NetWare for SAA эти пять сеансов могут относиться к нескольким главным системам.
- Программы SEND/RECEIVE или TRANSFER позволяют осуществлять обмен файлами между PC и главной ЭВМ.
- Совместим с программами управления памятью других поставщиков, например, программой Connecting/Room фирмы Helix.

- Поддерживает API, являющиеся промышленными стандартами, что предотвращает капитальные вложения пользователей в настоящем и будущем. Эти интерфейсы прикладного программирования включают EEHLLAPI и Low-Level API фирмы IBM, а также NetWare 3270 API фирмы Novell.
- Простая в использовании программа определения конфигурации использует IBM-совместимый интерфейс CUA.
- Утилита определения клавиатуры позволяет воспроизводить клавиатуру на экране, что упрощает переназначение функций клавиш, в том числе и "горячих".
- Поддерживает более 30 клавиатур с иностранными алфавитами.
- Поддерживает клавиатуру IBM с 83/84 клавишами и усовершенствованную со 101/102 клавишами, а также клавиатуры фирмы Key Tronic со 122 клавишами.
- Поддерживает перенаправление печати на экран и принтеры 3287 главной системы на принтеры сети.
- Поддерживает, расширенный поток данных, включая негативное видеоизображение, подчеркивание, подсветку и семь цветов.
- Пакет NetWare 3270 LAN Workstation for DOS может быть легко выгружен из резидентной памяти без необходимости перезагрузки рабочей станции, что позволяет пользователям выполнять другие прикладные программы.
- Обеспечивает средство внутреннего отслеживания для диагностики.
- Поддерживает световое перо и устройства, совместимые с мышью фирмы Microsoft, а также функции, поддерживаемые этими устройствами, например, выбор поля по табулятору.

### Реализация

NetWare 3270 LAN Workstation for DOS может эмулировать на одной рабочей станции до пяти сеансов диалога (LU2), печати (LU1 или LU3) и APPC (LU6.2). Эти LU должны быть определены в программе VTAM (выполняющейся на главной системе и управляющей сетью главной ЭВМ) и/или в программе Network Control Program (выполняющейся на фронтальном процессоре), прежде чем будет осуществлено соединение с главной системой. Супервизору сети, устанавливающему NetWare 3270 LAN Workstation for DOS может потребоваться помощь и консультация персонала, осуществляющего управление большой ЭВМ.

## NetWare 3270 LAN Workstation for Macintosh

NetWare 3270 LAN Workstation for Macintosh — это пакет эмуляции терминала IBM, использующийся вместе NetWare for SAA. Он обеспечивает пользователям Apple Macintosh, работающим в ЛВС NetWare, недорогой доступ к большим ЭВМ архитектуры IBM SNA и миниЭВМ AS/400 при помощи NetWare for SAA.

Являясь прикладной программой Macintosh, пакет NetWare 3270 LAN Workstation for Macintosh полностью интегрирован со средой Macintosh. Пользователи могут в полной мере использовать преимущества дружественного графического интерфейса Macintosh, включая интенсивное использование мыши, функций копирования и вставки, а также поддержку адаптации размеров шрифтов, воспроизведения клавиатуры на экране и различных цветовых палитр.

NetWare 3270 LAN Workstation for Macintosh поддерживает одновременно до 26 сеансов диалога с главной системой на одной рабочей станции. Пользователи, имеющие права доступа, могут устанавливать сеансы с несколькими главными системами SNA для доступа к данным и прикладным программам. Одна из опций меню обеспечивает передачу файлов с рабочей станции в главную ЭВМ и в обратном направлении. NetWare 3270 LAN Workstation for Macintosh продается как по лицензии на сервер, так и по лицензии на рабочую станцию.

### Характеристики

- Эмулирует терминал IBM 3270, поддерживая полноэкранные модели Model 2, 3, 4 и 5.
- Поддерживает до 26 сеансов диалога с главной системой на одной рабочей станции ЛВС.
- Обеспечивает передачу файлов с рабочей станции в главную ЭВМ и обратно, используя программу передачи файлов главной ЭВМ IBM IND$FILE в средах TSO, CMS и CICS.
- Поддерживает промышленный стандарт IBM 3270 EEHLLAPI.
- Обеспечивает интерфейс HyperCard к EEHLLAPI.
- Опция меню, предназначенная для графической разметки клавиатуры, обеспечивает вывод на экран клавиатуры, что позволяет легко переназначать функциональные клавиши.
- Обеспечивает исчезающие меню для воспроизведения клавиатуры.
- Обеспечивает исчезающую малую клавиатуру для удобства активизации функциональных клавиш главной системы при помощи мыши.
- Включает возможность распечатки окна диалога с главной системой.
- Поддерживает печать в формате PostScript через NetWare for Macintosh.

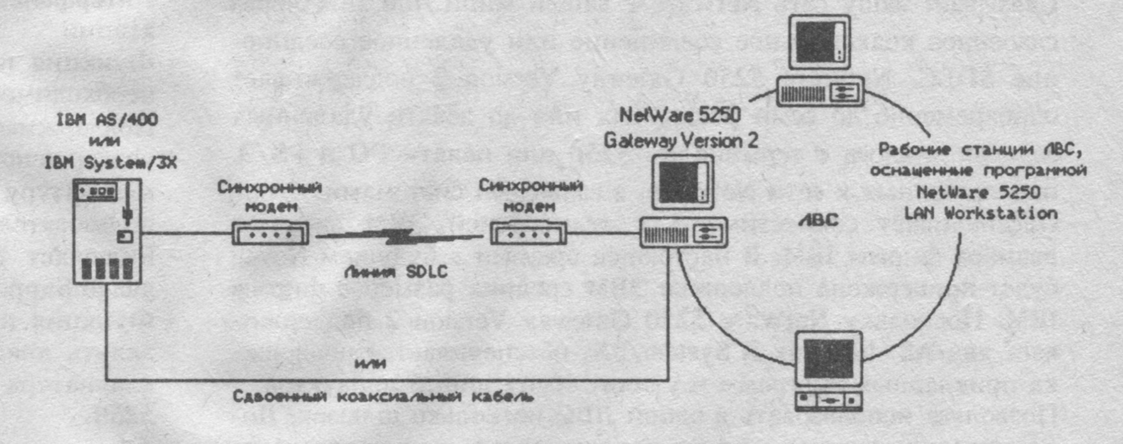



Рис. 3.35. NetWare 5250 Gateway Version 2

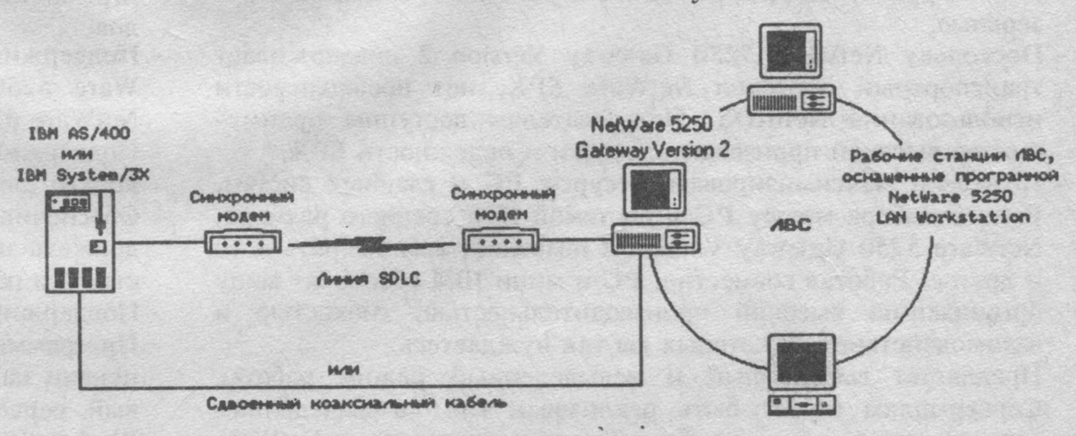



Рис. 3.36. NetWare 5250 LAN Workstation

- Поддерживает расширенный поток данных, включая негативное видеоизображение, подчеркивание, подсветку и семь цветов.
- Позволяет пользователям возможность изменять цвета и размер шрифтов в окнах сеанса с главной системой в соответствии с их индивидуальными привычками.
- Обеспечивает опцию автоматического выбора шрифтов в соответствии с размером окна сеанса с главной системой.
- Обеспечивает поддержку мыши для позиционирования курсора, операций копирования и вставки текста из окон диалога с главной системой и в них, а также эмуляцию светового пера. Совместим с операционной системой Macintosh, включая System 7.0.
- Включает протокол AppleTalk Data Stream Protocol (ADSP).

## NetWare 5250 Gateway Version 2

NetWare 5250 Twinax Gateway Version 2 превращает ПЭВМ PC или PS/2, работающие в сети NetWare, в сервер-шлюз, обеспечивающий связь с миниЭВМ IBM AS/400 или System/3X. Этот шлюз, осуществляющий связь с AS/400 или с System/3X через локальное сдвоенное коаксиальное соединение или удаленное соединение SDLC, позволяет IBM PC, PS/2 и совместимым ПЭВМ осуществлять доступ к информации и прикладным программам этих миниЭВМ.

NetWare 5250 Gateway Version 2 позволяет пользователям AS/400 или System/3X одновременно использовать функции, предоставляемые персональными компьютерами и ЭВМ среднего размера, что сокращает общие затраты на аппаратуру. NetWare 5250 Gateway Version 2 поддерживает одновременно до семи локальных или до девяти удаленных сеансов диалога или печати PC, подключенных к сети, с главными системами, причем пять из этих сеансов могут одновременно поддерживаться на одной рабочей станции.

### Характеристики

- Связывает вашу сеть NetWare с вашей миниЭВМ IBM через сдвоенное коаксиальное соединение или удаленное соединение SDLC. NetWare 5250 Gateway Version 2 поддерживает одновременно до семи локальных или до девяти удаленных сеансов диалога с терминалом 5250 или печати PC и PS/2, подключенных к сети NetWare, с главными системами.
- Обеспечивает совместимость с технологией ЭВМ среднего размера фирмы IBM. В настоящее время и в будущем Novell будет привержена поддержке ЭВМ средних размеров фирмы IBM. Поскольку NetWare 5250 Gateway Version 2 поддерживает как AS/400, так и System/3X, обеспечивается поддержка прикладных программ и утилит обеих линий продуктов.
- Позволяет использовать в одной ЛВС несколько шлюзов. Дополнительные шлюзы могут использоваться для распределения нагрузки, поддержки нескольких миниЭВМ и как резервные.
- Поскольку NetWare 5250 Gateway Version 2 поддерживает транспортный протокол NetWare SPX, нет необходимости использования NetBIOS. Пользователям доступны преимущества высокой производительности и надежности SPX.
- Позволяет максимизировать ресурсы PC и главных систем. Вместо выбора между PC и системой IBM среднего размера, NetWare 5250 Gateway Version 2 позволяет вам выбрать и то и другое. Работая совместно, PC и миниЭВМ обеспечат вашу организацию высокой производительностью, гибкостью и взаимодействием, в которых вы так нуждаетесь.
- Предлагает выделенный и невыделенный режим работы. Сервер-шлюз может быть реализован как на выделенном сервере, так и в виде комбинации сервера-шлюза и NetWare 5250 LAN Workstation. Выделенный режим обеспечивает оптимальную надежность и производительность. Невыделенный режим сокращает затраты на аппаратные средства.
- Работает вместе с NetWare 5250 LAN Workstation. Эта простая в использовании программа эмуляции терминала 5250 используется вместе с NetWare 5250 Gateway Version 2 для обеспечения связи между ЛВС и миниЭВМ.

## NetWare 5250 LAN Workstation

NetWare 5250 LAN Workstation — это программа эмуляции терминала IBM, используемая вместе с NetWare 5250 Gateway Version 2. Продукт обеспечивает PC, PS/2 и совместимым ПЭВМ недорогой доступ к миниЭВМ IBM AS/400 или System/3X.

Программа NetWare 5250 LAN Workstation обеспечивает поддержку нескольких сеансов и разнообразные возможности, имеющиеся в реальных устройствах 5250. Она поддерживает до пяти сеансов терминалов и принтеров с главной системой на одной рабочей станции, передачу управления в DOS и из DOS по "горячей" клавише, а также обычные операции сеанса DOS.

### Характеристики

- Обеспечивает эмуляцию устройств IBM AS/400 и System/3X различных типов. Поддерживает терминалы IBM 3180, 3196, 5251, 5291 и 5292 и принтеры IBM 3812, 5219, 5224, 5225 и 5256.
- Поддерживает множество прикладных программ AS/400 и System/3X.
- При совместном использовании с NetWare 5250 Gateway Version 2 поддерживает до семи локальных и до девяти удаленных сеансов с главными системами и один сеанс DOS на одной рабочей станции.
- Интерфейс меню обеспечивает простоту установки и эксплуатации.
- Функция помощи, чувствительная к контексту, уменьшает необходимость сопровождения.
- Поддерживает стандартную клавиатуру с 83/84 клавишами, усовершенствованную клавиатуру со 101/102 клавишами и клавиатуру фирмы Key Tronic со 122 клавишами, что дает пользователю свободу выбора клавиатуры.
- Включает поддержку национальных языков National Language Support для более чем 10 языков.
- Функция переопределения клавиатуры позволяет переопределить некоторые клавиши. По желанию пользователя вся клавиатура может быть легко переопределена под схему 5250.
- Обеспечивается реальная возможность запоминания и "проигрывания" часто используемых последовательностей символов.
- Поддерживает передачу файлов, используя программу NetWare 5250 File Transfer, входящую в комплект поставки NetWare 5250 LAN Workstation.
- Поддерживает широкоэкранный режим дисплея (132 колонки) для рекомендованных видеоадаптеров и мониторов.
- Обеспечивает 128-символьный буфер, позволяющий продолжать набор символов, не дожидаясь ответа главной системы и обновления экрана.
- Поддерживает IBM PC Support/36 и PC Support/38.
- Программа NetWare 5250 LAN Workstation продается по лицензии на один сервер и может быть загружена на файловый сервер или отдельную рабочую станцию 5250 LAN Workstation.

*Новость месяца и одна из новостей года: фирма Hewlett-Packard выпустила новое поколение принтеров. 26 октября было объявлено о начале продажи двух новых принтеров. Это LaserJet 4 и LaserJet 4M.*

# Хит сезона от модельеров Hewlett-Packard

Итак, очередное поколение принтеров стартовало. Что же нового предлагает фирма на этот раз?

Нового много, совершенно нового нет — все это уже было в той или иной форме, но никогда еще не собиралось вместе в одном устройстве. Первое и главное: разрешающая способность принтера составляет 600x600 точек на дюйм при полной совместимости с режимом 300 точек. Это одна из трех мер по повышению качества печати, отличающих LaserJet 4. (Обратите внимание на написание нового названия! Предыдущие принтеры обозначались римскими цифрами — LaserJet II, LaserJet III, LaserJet IIIP, — а это поколение обозначено арабской — LaserJet 4.) Две другие — это использование фирменной технологии улучшения разрешения RET и применение нового тонера с уменьшенными частицами. Использование RET делает качество печати сравнимым с качеством принтеров с разрешением 800-900 dpi.

Понятно, что принтер с удвоенным разрешением при построении страницы обрабатывает вчетверо больше данных, но LaserJet 4 работает не медленнее своих собратьев — его контроллер построен на мощном 20-мегагерцевом RISC-процессоре Intel 960. Он позволяет обеспечить быструю обработку данных и их быструю передачу через интерфейсы принтера. Первая страница печатается через 21 секунду после старта, а скорость работы нового привода печати составляет 8 страниц в минуту.

**Основные параметры** *HP LaserJet 4*

| | |
|---|---|
| **Разрешение** | 600x600 dpi |
| **Скорость печати** | 8 стр/мин |
| **Время печати 1-ой страницы** | 21 с |
| **Объем ОЗУ** | 2 Мбайта (станд.) до 34 Мбайт |
| **Язык** | улучшенный PCL 5 |
| **Ресурс печати** | 20000 стр/мес |
| **Размеры** | 416x403x297 мм |
| **Масса** | 16.8 кг |

В LaserJet 4 применен несколько усложненный двунаправленный интерфейс Bi-Tronics. Теперь не только компьютер управляет принтером, но и принтер может управлять драйвером печати, делая работу всей системы более эффективной. Кроме всего прочего, это позволяет не подходя к принтеру осуществлять управление им и проверку его состояния с компьютера. Предусмотрен слот для установки сетевой платы, превращающей принтер в сервер печати. Принтер автоматически переключается между портами, определяя активный. Таким образом к нему можно подключить: по одному компьютеру через параллельный и последовательный порты, и многомного машин через сеть.

В принтер встроена улучшенная версия языка управления страницами PCL 5, отличающаяся от предыдущей ускоренной векторной графикой, улучшенной технологией сжатия данных и увеличенным количеством резидентных шрифтов (что, впрочем, отечественного пользователя совершенно не греет — эти шрифты не содержат русских символов). Особо нужно отметить технологию сжатия. При обычной технологии на стандартную страницу, напечатанную с разрешением 600x600 точек на дюйм, требуется около 6 Мбайт памяти. LaserJet 4 обходится двумя, печатая страницу на формате Legal, площадь которого на 25% больше.

Фирма поставляет интерпретатор языка Adobe PostScript Level II, выполненный в виде SIMM-модуля. При его установке принтер автоматически распознает входной язык и использует соответствующий интерпретатор.

В этой модели устранены недостатки предыдущих моделей, вызывавшие много нареканий. Значительно улучшена система подачи бумаги — у LaserJet 4 нет огромного подающего лотка, торчащего на двадцать с лишним сантиметров, здесь два основных источника бумаги: во-первых, кассета с 250 листами, смонтированная в нижней части принтера, во-вторых, удобный и прочный откидывающийся наклонный лоток на 100 листов (примерно как в LJ IIP или LJ IIIP, но значительно аккуратнее). Кроме того, можно установить податчик конвертов (их пять типов) и дополнительную кассету для 500 листов бумаги; в последнем случае принтер может одновременно "запасти" 250+100+500=850 листов бумаги! Это отлично, особенно для работы в сети.

Уменьшены габариты принтера и потребляемая мощность, печатает он достаточно тихо (49 дБ), а в спокойном режиме его вообще не слышно (33 дБ). Прогревается принтер за 30 секунд, что вдвое меньше аналогичного показателя для предыдущих моделей. Появилось много приятных мелочей, облегчающих работу или делающих ее более комфортной. Стоимость этого принтера ровно такая же, как и у LaserJet III — 2785 долларов. Гарантийный срок — 1 год.

Одновременно выпускается версия для работы в многосредных системах, названная LaserJet 4M. Этот принтер отличается от базового варианта наличием встроенного интерпретатора PostScript Level II, стандартным ОЗУ объемом 6 Мбайт, дополнительным интерфейсом LocalTalk. Этот принтер предназначен для одновременной работы с компьютерами семейств IBM PC и Apple Macintosh, с языками PCL и PostScript. Кроме того, с ним может поставляться картридж с русскими шрифтами. LaserJet 4M стоит 3621 доллар.

И хотя новый принтер фирмы Hewlett-Packard не является революционным (уже давно выпускаются аппараты с разрешением 1000 и 1200 dpi, спецплаты на могучих RISC-процессорах, обмолачивающие PostScript-страницу за полминуты и автоматически переключающиеся на PCL), а IBM уже больше года продает аналогичный принтер, имеющий даже чуть лучшие показатели, у LaserJet 4 большое будущее. Просто потому, что это Hewlett-Packard.

*И.Вязаничев*




# СП "МИКРОИНФОРМ"

*От Москвы до самых до окраин,*
*С южных гор до северных морей,*
*ЛЕКСИКОН — известный всем редактор*
*В необъятной Родине моей...*

**Самый распространенный отечественный программный продукт — многооконный текстовый процессор ЛЕКСИКОН — идеальное средство для обработки документов на русском, английском и еще многих языках (испанском, немецком, французском, украинском, казахском, армянском, грузинском, латышском и др.).**

**ЛЕКСИКОН позволяет:**

- вводить и редактировать текст на русском, английском и других языках;
- работать с несколькими документами в разных окнах (до десяти);
- проводить контекстный поиск и замену;
- форматировать абзацы и страницы;
- автоматически составлять оглавление;
- осуществлять автоматический перенос слов по слогам;
- проверять правильность написания слов по правилам русской орфографии;
- использовать макрокоманды для ускорения повторяющихся операций;
- конвертировать текст в форматы **XEROX VENTURA PUBLISHER** и **MS WORD.**

Мы всегда рады видеть Вас

# СП "МИКРОИНФОРМ"

*113184 Москва, ул.Островского, 44*
*Тел. (7-095) 233-00-06*
*Факс (7-095) 235-10-53*
*Телекс 411660 MICRO*
*E-mail: lexicon@micro.msk.su*




*В этой статье мы представляем новые компьютеры фирмы NEC Technologies, которая является подразделением крупнейшей корпорации NEC. Во всем мире эта фирма известна, по крайней мере, как изготовитель отличных мониторов серии MultiSync. Однако не менее известна и другая ее продукция. Это лазерные принтеры SilenWriter, цветные принтеры ColorMate, лэптопы Prospeed, ноутбуки UltraLite. Сегодня мы расскажем о новых настольных компьютерах PowerMate серии Image, которые также выпускаются фирмой NEC Technologies.*

# Компьютеры PowerMate

В мае этого года фирма NEC Technologies объявила о пополнении семейства настольных компьютеров PowerMate серии Image новыми моделями: SX/25i, 486/33i и 486/50i. Компьютеры PowerMate были впервые анонсированы в октябре 1991 года. До недавнего времени в это семейство входили следующие модели компьютеров: SX/16i, SX/20, SX/20vi и 386/33i.

Одной из отличительных особенностей компьютеров PowerMate является возможность наращивания их вычислительной мощности (upgrade) путем простой замены системной платы. Однако, как заметил на презентации директор по маркетингу фирмы NEC Technologies Денни Лэйн (Denny Lane), "легкое наращивание вычислительной мощности — это не единственное существенное отличие серии компьютеров Image". Не менее важным отличием компьютеров PowerMate является наличие локальной видеошины (local video bus) графической подсистемы. Фирма NEC Technologies была одной из первых фирм, которая стала использовать в своих компьютерах видеошину как стандартную особенность.

Использование отдельной видеошины позволяет существенно повысить производительность компьютера при работе с графикой, без применения специальных видеоадаптеров с аппаратными ускорителями вычислений. Благодаря наличию отдельной видеошины производительность графической подсистемы по сравнению со стандартным VGA-адаптером возрастает почти в 4 раза. Объясняется это достаточно просто.

Скорость передачи информации между центральным процессором и видеоадаптером ограничивается максимальной тактовой частотой системной шины (обычно ISA), частота которой, как правило, не превосходит 8 МГц. Это, по сути, обуславливает узкое место любой высокопроизводительной системы, работающей на тактовой частоте 20 и более МГц. Локальная видеошина, предложенная фирмой NEC Technologies, позволяет передавать по ней данные с той же скоростью, с какой работает сам процессор. Например, для компьютера PowerMate SX/25i тактовая частота видеошины также составит 25 МГц. Отметим, что видеоадаптер в компьютерах серии Image интегрирован на системной плате и оснащен videoRAM размером 1 Мбайт.

Другой отличительной особенностью компьютеров PowerMate является использование технологии ImageSync, которая позволяет распознавать подключение к видеоадаптеру этих компьютеров мониторов серии MultiSync FG. Если такое подключение имеет место, то видеоадаптер настраивается на самую высокую частоту кадровой развертки. В результате изображение на экране монитора становится более отчетливым и практически не мерцает.

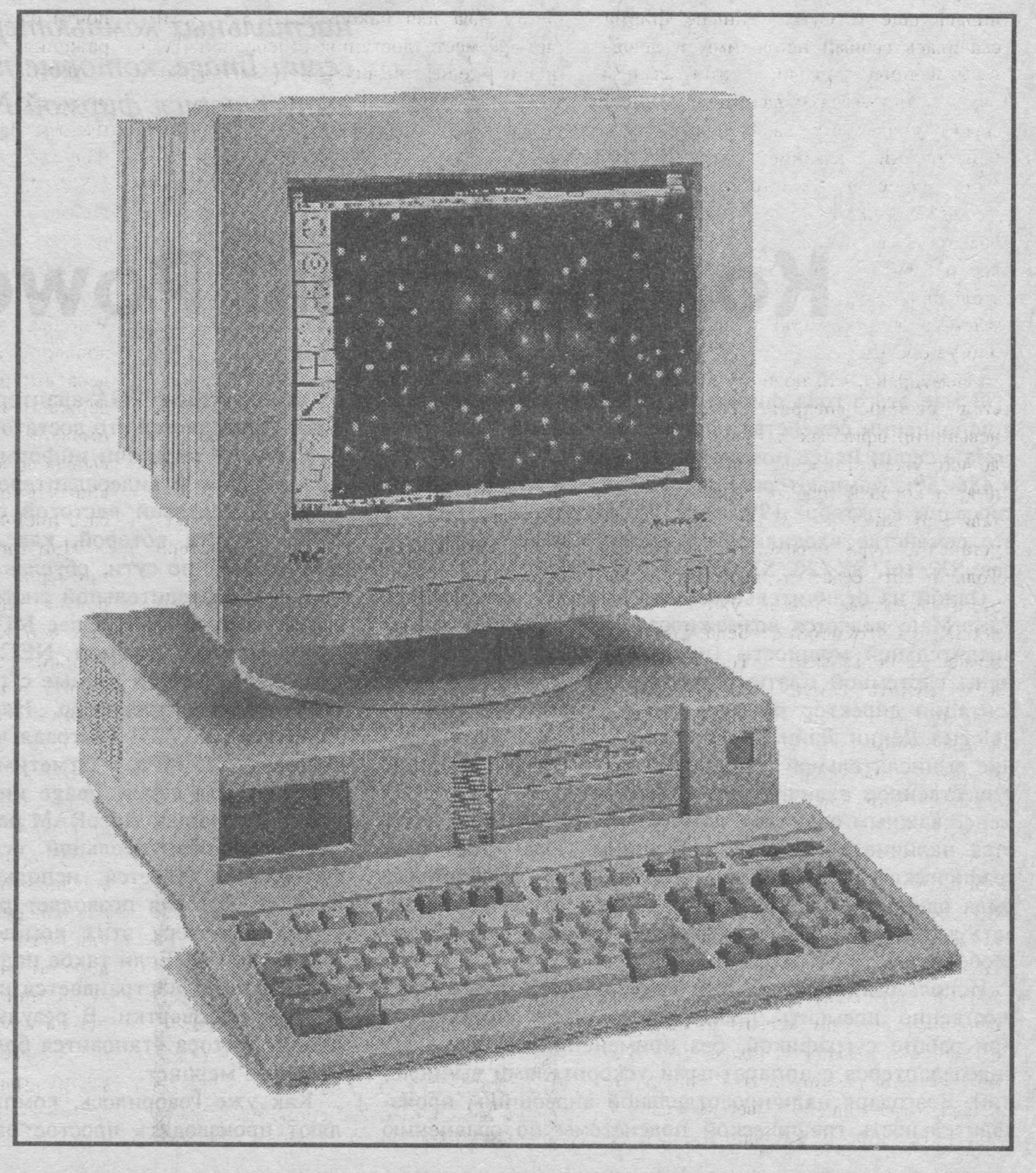
Как уже говорилось, компьютеры PowerMate позволяют производить простое наращивание вычислитель-

ной мощности путем замены системной платы. Таким образом, процессор, кэш, шина памяти и видеоподсистема заменяются одновременно. У модели PowerMate 486/33i на системной плате находится также специальное гнездо OverDrive, которое может быть использовано для установки процессоров, работающих на удвоенной частоте (i486DX2-50, i486DX2-66). Компьютер PowerMate 486/50i базируется именно на таком процессоре. На системной плате компьютеров может быть установлено до 36 Мбайт оперативной памяти.

Все новые компьютеры серии Image оснащены миниатюрными винчестерами и приводами флоппи-дисков типа SlimLine (1 дюйм по высоте). Емкость таких винчестеров составляет 120 или 240 Мбайт, в зависимости от спецификации. Привод для флоппи-диска размером 3,5 дюйма входит в стандартную конфигурацию компьютеров, а для размера 5,25 дюйма является факультативным. В корпусе компьютеров PowerMate предусмотрено 4 свободных места для установки дисководов, а на системной плате имеется 4 свободных 8/16-разрядных слота расширения, соответствующих стандарту ISA. Максимальный размер кэш-памяти для моделей SX/25i и 486/33i составляет соответственно 16 и 128 Кбайт. Размер кэш-памяти второго уровня в модели 486/50i равен 128 Кбайт (как известно, процессор i486 имеет также 8-Кбайтный внутренний кэш). Безусловным новшеством компьютеров PowerMate 386/25i является 128-Кбайтный ROM BIOS, выполненный на микросхемах NEC FLASH, что позволяет обновлять содержимое BIOS непосредственно со специальной дискеты. Все модели имеют один (или два) последовательный и один параллельный порты, а также порты для подключения клавиатуры и мыши (стиль PS/2 — 6 контактов). Пока несколько необычным кажется коннектор для мультимедиа, используемый для подключения микрофона и головных телефонов (наушников). Особо хотелось бы отметить высокую степень системной защищенности компьютеров PowerMate. Отдельная парольная защита предусмотрена при включении питания, для работы с клавиатурой и при использовании компьютера в режиме сервера. Имеется возможность защиты от загрузки с дискеты и записи на нее, а также управление режимами работы последовательного и параллельного портов. Ключевая защита может использоваться и на крышке самого компьютера.

Все компьютеры серии Image поставляются со стандартным программным обеспечением, в которое входит MS-DOS 5.0 и Windows 3.1. Преимущества компьютеров PowerMate особенно ярко проявляются при их использовании в настольных издательских системах и при работе с Windows.

*А.Борзенко*



# ZEOS 486SLC-25 – КАЧЕСТВО НОВОЕ, ЦЕНА СТАРАЯ

Процессоры для компьютеров выпускают все больше и больше фирм: сейчас, кроме доминирующей на этом рынке Intel, на нем активно работают такие фирмы, как AMD, Chips&Technologies, NEC. Теперь к ним присоединилась еще и Cyrix. Раньше фирма славилась своими неплохими и дешевыми сопроцессорами, а этой весной она заметно встряхнула рынок процессоров, представив свой новый чип, получивший название Cyrix 486SLC. Этот процессор — нечто среднее между 386SX и 486SX; в нем использована архитектура, близкая к 486-м процессорам, хотя сто́ит он практически столько же, сколько Intel 386SX-25, и использует обычную 16-разрядную шину, как и 386SX.

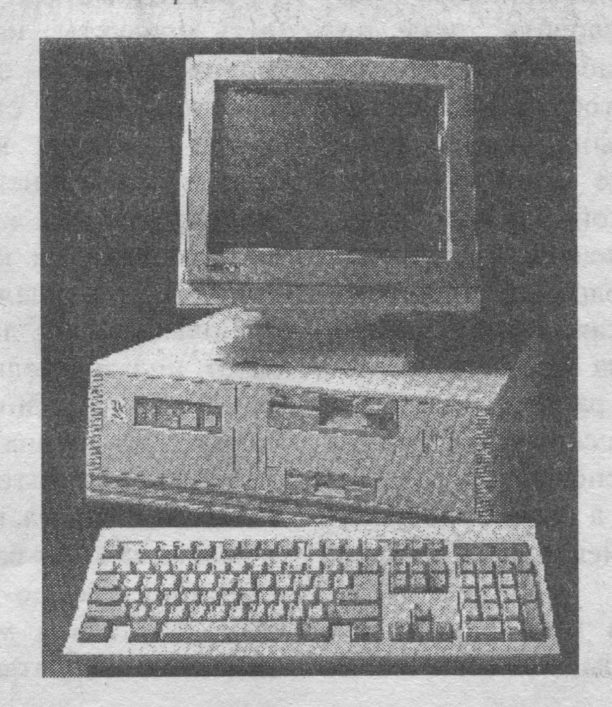
Немудрено, что новинку заметили и стали активно внедрять. Так появился новый тип офисных компьютеров, после чего уже стала совсем явной тенденция, о которой любят говорить представители Intel, — 386-е процессоры устарели, пора переходить на 486-е. Только вот беда: первый процессор этого семейства нынче уже не Intel 486SX, а Cyrix 486SLC! Беда для Intel, но не для пользователя. Теперь вы можете купить самую простую 486-ю машину за те деньги, которые совсем недавно пришлось бы выложить за такую же, но на 80386SX-20.

Одна из первых машин на этом процессоре — это ZEOS 486SLC-25, появившаяся в начале апреля. Очень неплохой конфигурации (4 Мбайта ОЗУ, 107 Мбайт винчестер, два флоппи-диска, видеоадаптер HiColor VGA 1024x768x32768 цветов, цветной монитор 14“, комплект программного обеспечения), она стоит всего 1,995 долл.

Машина обладает большими возможностями расширения — на плате предусмотрено целых восемь слотов. Один из них 8-разрядный, остальные — 16-разрядные. Только один 16-разрядный слот занят платой видеоадаптера, остальные семь свободны. Система имеет семь мест для установки дисковых накопителей — два половинной высоты под накопители 5.25“, два для накопителей 3.5“ (эти четыре места доступны с передней панели) и еще три внутренних места для установки 5.25-дюймовых накопителей. Система поставляется с двумя гибкими дисками (5.25/1.2 и 3.5/1.44) и IDE-винчестером Seagate ST3120A емкостью 107 Мбайт и скоростью доступа 18 мс. Добавив 210 долларов, можно заменить этот винчестер на более быстрый и емкий ST3283A (245 Мбайт/12 мс).

На материнской плате размещены два последовательных порта, параллельный и порты для гибких дисков и IDE-винчестеров. Интересная подробность: оперативность выпуска этого компьютера объясняется тем, что его плата представляет собой усовершенствованный вариант платы компьютера ZEOS 386SX-20.

Компьютер оснащен видеоадаптером Dimond SpeedStar HiColor с 1 Мбайтом видео-ОЗУ, который обеспечивает работу с максимальным разрешением 1024x768 без чередования; в комплект входит цветной 14-дюймовый VGA-монитор. Видеосистема обеспечивает воспроизведение 32768 цветов, что дает почти фотографическое качество изображения.



Фирма ZEOS позаботилась о своих покупателях и снабдила компьютер неплохим набором программного обеспечения: стандартно поставляется MS-DOS 5.0, Windows 3.1 с Microsoft Mouse, кроме того, можно выбрать два из трех пакетов: 1-2-3 for Windows, Ami Pro 2.0 или Freelance for Windows. Если вы намерены активно работать с Windows, имеет смысл добавить сотню долларов и оснастить компьютер видеоадаптером Dimond Stealth VGA, содержащим ускоритель графики и имеющим скорость работы, вдвое превышающую среднюю скорость работы видеосистем более мощных машин на базе Intel 486SX-25.

При той же цене, ZEOS 486SLC-25 примерно на 45% быстрее компьютеров на 386SX-25, но все-таки медленнее 486SX-25. Однако для работы с Windows эта машина подходит как нельзя лучше — ее производительность заметно выше, чем у конкурентов.

Одно дополнение — пока что не сняты все опасения, касающиеся совместимости нового процессора с процессорами Intel, хотя все тесты проходят успешно и пока еще не поступало сообщений о проблемах с ним. Учитывая опасения пользователей, ZEOS обещает заменить ваш компьютер на более дорогой 486SX-25 в конфигурации, аналогичной вашей, если в течение полугода вам удастся обнаружить, что на нем не работает любая из 100 наиболее распространенных программ.

*И.Вязаничев*



# Обслуживание лазерных принтеров

Пожалуй, из всех видов стандартного оборудования, имеющего какое-либо отношение к компьютерам, лазерный принтер может создать наибольшее количество проблем. Вследствие сложности конструкции и наличия высокочувствительных прецизионных узлов он подвергается механическим повреждениям чаще, чем другие периферийные устройства. Кроме того, лазерный принтер подвержен влиянию внешней среды. Такие, казалось бы, мелочи, как пыль, дым, мусор с поврежденной бумаги, даже шерстинка вашего любимого кота, могут, попав внутрь принтера, вызвать массу проблем — от неприятностей с подачей бумаги до понижения качества печати.

Большинство изготовителей советуют владельцам лазерного принтера проверять его у специалиста после печати 50-100 тыс. страниц. В остальное время вы можете сами позаботиться о поддержании принтера в хорошем состоянии. Например, производить такие несложные операции, как замена барабанов и кассет с тонером, а также следить за тем, чтобы в машине не скапливались грязь и частички пыли, попадающие внутрь при работе принтера, вполне можно самостоятельно. Это уже будет предпосылкой для его безотказной работы и хорошего качества печати.

Если вы никогда не занимались подобной профилактикой и не производили замены каких-либо частей принтера, то, вероятно, будете весьма удивлены легкостью этих операций. Поддержка работы лазерного принтера не потребует от вас колоссальных научных знаний. Достаточно твердой руки и способности разбираться в рисунках и инструкциях, а также добросовестного отношения к делу и элементарного набора домашних инструментов.

Каждый производитель принтеров рекомендует для своего изделия различные процедуры по обслуживанию. Справочный материал и инструкции, которыми сопровождается каждая заменяемая деталь, являются лучшими источниками достаточно подробной информации о том, как поддерживать нормальную работу именно вашего принтера. Здесь я дам лишь общее представление о механизме работы лазерного принтера, поговорив о двух основных типах конструкций лазерных принтеров и требованиях по их обслуживанию, а также расскажу, как заменить тонер и барабан. Эти рекомендации помогут осуществлять регламентное обслуживание вашего принтера.

## Чем отличается обслуживание разных моделей принтеров

Расходные материалы для лазерного принтера, то есть продукты, используемые для печати и регулярно пополняемые — это светочувствительный барабан, на котором лазер строит изображение, порошок-тонер, “прилипающий“ к странице и создающий окончательное изображение, в некоторых моделях — проявляющий порошок, который способствует передаче тонера на бумагу, еще нужно бывает менять отдельный чистящий блок, удаляющий остатки тонера с барабана.

Большинство лазерных принтеров так или иначе дают пользователю знать, когда пора заняться обслуживанием. В одних моделях имеется сигнальная лампочка, загорающаяся, когда появляется необходимость заменить эти материалы; другие принтеры имеют окно, изменение цвета которого также указывает на необходимость замены барабана и кассеты с тонером. Некоторые принтеры по-прежнему снабжены счетчиком страниц; последние модели обычно пишут руководство к действию открытым текстом на своем маленьком жид-

кокристаллическом дисплее. И все-таки самым явным показателем того, что пришло время позаботиться о принтере, является качество отпечатков.

На рынке преобладают два основных типа лазерных принтеров. Первый тип предполагает объединение всех требующих замены материалов в едином картридже. Примером таких устройств могут служить лазерные принтеры: Hewlett Packard LaserJet Series II, использующий привод Canon SX; HP LaserJet IIP, использующий привод Canon LBP-LX, а также Canon LBP-4 с приводом Canon-P110. Когда машина такого типа нуждается в тонере (обычно после того, как напечатает 4-5 тыс. страниц), вы вытаскиваете и заменяете картридж, то есть и барабан, и тонер, и чистящий блок сразу.

Второй тип принтеров предполагает раздельные картриджи для расходуемых материалов. Примерами этого типа машин являются EPL-6000 фирмы Epson и SHARP JX-9500 фирмы SHARP. Привод первого принтера создан фирмой Tokio Electric и рассчитан на печать приблизительно 300 тыс. страниц, барабана хватает на печать 10 тыс. страниц, зато кассета с тонером требует замены после каждых 1.5 тыс. отпечатков. Второй принтер меньше нуждается в обслуживании — барабан меняется после печати 50,000 страниц, а тонера хватает аж на 30 тыс. листов.

Замена "единого" картриджа производится легко и быстро, занимая от силы четверть часа. По существу, этот процесс заключается в извлечении старого картриджа и установке нового. Когда вы распаковываете новый картридж, внимательно и точно следуйте приведенным в инструкции изготовителя рекомендациям. Перед тем как установить новый картридж, обычно рекомендуется его раз десять встряхнуть, чтобы тонер внутри распределился равномерно. Сама установка обычно не вызывает проблем — просто аккуратно установите новый картридж на то место, откуда только что был извлечен старый. Если вы немного поспешили и захлопнули крышку принтера сразу после установки картриджа, вы будете удивлены тем, что он откажется печатать. Вы бы на его месте поступили точно так же. Одно важное правило при установке нового картриджа: не забывайте выдергивать из него страховочную ленту, предотвращающую рассыпание тонера при транспортировке. "Хваталка" для выполнения этой нехитрой процедуры находится сбоку картриджа. Уверенно, но без рывков тащите ее до тех пор, пока лента не будет извлечена полностью. При этом она должна оторваться; теперь можно опустить ее в мусорную корзину.

В машинах, где барабан и кассета с тонером разделены, во время замены тонера есть шанс сильно перепачкаться, поэтому для этой операции рекомендуется приготовить пару старых газет или несколько листков ненужной бумаги. После того как вы откроете крышку и вытащите резервуар с тонером (в некоторых моделях называемый корзинкой) из машины, положите его на заранее приготовленные газеты. Перед удалением кассеты с израсходованным тонером (точнее, уже без него) полезно постучать по ней по всей длине. Дело в том, что тонер является довольно липким порошком и некоторое его количество остается на стенках кассеты. Вытряхнув его, вы сэкономите несколько центов и уменьшите количество грязи на столе. Теперь ставьте новую кассету, затем аккуратно извлекайте страховочную пластиковую пленку, которая предохраняет тонер внутри картриджа.

Замена барабана в аппаратах данного типа является более сложной операцией. Процедура ее выполнения сильно зависит от марки принтера, поэтому для того, чтобы сделать все правильно, предварительно внимательно ознакомьтесь с инструкциями, прилагаемыми к каждому барабану. Не забывайте и о том, что распакованный барабан можно подвергать воздействию света только в течение очень короткого промежутка времени; не стоит работать на солнце — барабан может долго приходить в себя после такой экзекуции, наказав вас посредственным качеством печати и пятнами и полоса-

## Как работает лазерный принтер

В большинстве лазерных принтеров используется маломощный полупроводниковый лазер, формирующий изображение на светочувствительном фотоприемном барабане. Барабану предварительно сообщается статический заряд. Когда луч лазера попадает на барабан, заряд стекает с поверхности. Освещаемые лазером участки барабана притягивают (или не притягивают — в другом варианте построения изображения) частицы порошкообразного тонера и те пристают к барабану в нужных местах, создавая изображение на нем.

Когда изображение на барабане построено и он покрыт тонером, подающий механизм берет лист бумаги и отправляет его в принтер. Дальнейшие процедуры сокрыты от глаз пользователя внутри принтера. Лист заряжается таким образом, чтобы тонер с барабана притягивался к бумаге, затем он проводится под барабаном. После этого изображение в виде частичек тонера оказывается на бумаге. Теперь дело за малым — закрепить на бумаге легкий порошок, улетающий при слабом дуновении. Фиксирующий узел прогревает тонер до температуры плавления, а специальные резиновые валики прижимают расплавленный тонер к бумаге. Комбинация нагревания-прижатия позволяет намертво вваривать тонер в бумагу для получения прочного изображения. Что написано лазерным принтером — удаляется только автогеном.

ми на изображении. Кроме того, барабан нельзя класть на поверхности, которые могут вызвать его повреждение (лучше всего его вообще не класть, а сразу установить на место). Царапины и сбои на барабане будут точно повторены на отпечатанной страничке. Нельзя также прикасаться к светочувствительным цилиндрам: тем самым вы можете надолго вывести их из строя.

### Очистка....

В наиболее распространенных типах лазерных принтеров перед каждой заменой тонера нужно производить чистку фиксирующего узла. Раскаленные частицы тонера могут скапливаться в этой части механизма и весьма неблагоприятно воздействовать на работу. Поэтому для многих типов привода в каждый новый комплект картриджа с тонером входит специальный чистящий блок.

Некоторые части принтера (особенно фиксирующий узел) разогреваются во время использования. Поэтому, кроме случаев, особо оговоренных в инструкции, производить операции по обслуживанию следует с выключенным принтером. Иными словами, вам следует отключить принтер и дать ему остыть, так как в противном случае вы просто-напросто рискуете обжечься.

Когда фиксирующий узел станет достаточно холодным для прикосновения, откройте дверцу и аккуратно извлеките старый чистящий блок. При этом следует держать его за места, покрытые термоизолирующим ворсистым материалом (обычно зеленого цвета). На одном конце новой прокладки имеется небольшой кусочек войлока, закрепленный липкой лентой. Вставьте этот конец прокладки в фиксирующий узел — туда, где устанавливается чистящий блок, — и протрите им фиксирующий валик. Закончив эту процедуру, вставьте прокладку на место и закройте крышку узла.

### Очистка внутренних частей механизма

Благоразумно производить очистку внутренних механизмов принтера одновременно с заменой барабана и тонера. Прежде чем приступить к этой процедуре, приготовьте следующие материалы: чистую сухую тряпку и сосуд с водой, несколько ватных тампонов и ручной электрический фонарик. Осмотрите ваш принтер и сравните расположение внутренних деталей с изображенным на схеме, которую следует все время держать перед глазами. Определите местонахождение тех частей, которые вы намереваетесь чистить и, если барабан и тонер входят в единый картридж, найдите внутри принтера небольшую чистящую щетку — обычно она крепится где-то под рукой, но в не слишком заметном месте.

Очистите все направляющие детали, пройдясь по всему тракту подачи бумаги. Делать это нужно тряпкой, влажной настолько, чтобы к ней приставали посторонние частички и остатки бумаги. Конечно, не стоит мочить ее, словно вы моете пол — принтеру может не понравиться освежающий душ. Затем найдите заряжающий бумагу узел — он состоит из коронирующего провода и длинного ряда маленьких коротких иголочек (это выглядит как расческа зубьями кверху, закрытая с двух сторон планками); иногда в этот узел входит еще несколько проволок, натянутых веером над основным коронирующим проводом. Очистите иголочки чистящей щеткой (или просто маленькой жесткой щеточкой).

Очистку главного коронирующего провода труднее производить в моделях принтеров с раздельным расположением барабана и тонер-картриджа, нежели в конструкциях с единым картриджем. И в том, и в другом типе принтеров этот провод находится вблизи поверхности барабана. В случае с единым картриджем он меняется после печати каждых 4000 страниц. В принципе, обычно не возникает необходимости чистить его, но если вы считаете, что это необходимо, используйте для этой цели обратную сторону щетки, входящей в комплект принтера. (На одном конце ее есть скошенное заострение с замшевым покрытием, которое можно вставить в специальную прорезь картриджа.) В машинах с раздельным расположением барабана и тонера главный коронирующий провод натянут вдоль барабана, и до него можно добраться через специальный паз в держателе барабана. При помощи ватного тампона, слегка нажимая, удалите грязь с провода. Но будьте осторожны, не оборвите его! Это грозит большими неприятностями — принтер просто перестанет печатать, а на бумаге останется грязь.

Затем проверьте все внутренние части принтера. При этом может оказаться полезным ручной фонарик. Выдвиньте лоток для бумаги и посмотрите, нет ли там обрывков или мелких частичек бумаги, пыли, волос и тому подобного мусора, который может прилипнуть к бумаге при работе.

Некоторые производители считают, что ко многим узлам принтера не следует прикасаться руками, можно



ИНТЕГРАТОР "ВИКТОРИЯ"

**Фирма ИнФоС представляет официальных дилеров**

## РОССИЯ

**Москва**
*СП "Аквариус Системз Интеграл"*
*Телефон: (095) 906-00-88*

*НПМП "Инвюрком"*
*Телефон: (095) 261-96-69*

**Новгород,** Колмово
*Фирма "Мединфо"*
*Телефон: (816-22) 2-37-88, 2-51-87*
*Факс: (816-22) 2-52-62*

**Новочеркасск,** Ростовская обл.
*Фирма "Иннотех"*
*Телефон: (863-52) 2-75-19, 2-30-84*
*Факс: (863-52) 2-33-24*

*МП "Инис"*
*Телефон: (863-52) 2-76-68, 2-46-59*
*Факс: (863-52) 3-04-42*

**Ковров,** Владимирская обл.
*Фирма "Синпро"*
*Телефон: (092-32) 3-21-22*

**Норильск**
*МП "Аквилон"*
*Телефон: (391-52) 4-00-10*

**Лысьва,** Пермская обл.
*Диагностический Центр*
*Телефон: (342-49) 2-07-87*

## УКРАИНА

**Павлоград,** Днепропетровская обл.
*Многоотраслевая фирма "Градиент"*
*Телефон: (056-72) 6-48-31*

**Снежное,** Донецкая обл.
*ИВЦ "Торезантрацит"*
*Телефон: (256-22) 5-41-80*

**Севастополь,** Крым
*Севастопольский Деловой Дом*
*Телефон: (0690) 52-92-83, 57-59-72*

## БЕЛАРУСЬ

**Минск**
*МГП "Норокон"*
*Телефон: (0172) 30-64-19*

## КАЗАХСТАН

**Кентау,** Шимкентская обл.
*Фирма "Жорга"*
*Телефон: (325-36) 3-39-91*

## УЗБЕКИСТАН

**Ташкент**
*Фирма "Информпроспект"*
*Телефон: (3712) 45-85-93, 56-57-97*

*ПРИГЛАШАЕМ ДИЛЕРОВ*

*Телефоны для справок:*
*(095) 471-32-63, (084-39) 2-24-82*
*Адрес: 113093 Москва, а/я 37*

## Решение обычных проблем печати

**1. Изображение слишком темное.**

Уменьшите плотность печати расположенным внутри принтера регулятором (если таковой имеется). Если регулятора нет — прочтите документацию, если ответа нет и там, позвоните на фирму, поставившую вам принтер.

**2. Изображение слишком бледное или не появляется вовсе.**

Проверьте уровень тонера. Посмотрите, зажегся ли предостерегающий сигнал. Если уровень тонера достаточен, легонько постучите по всей длине картриджа с тонером. Извлеките картридж (если вы работаете с принтером, построенном на приводе Canon) и покачайте его из стороны в сторону, несильно встряхните пару раз.

Увеличьте плотность печати регулятором внутри принтера.

Проверьте еще раз, правильную ли бумагу вы используете для вашего принтера.

**3. Что делать, если появились белые вертикальные линии или непропечатанные полосы.**

"Простучите" тонер-картридж по всей длине, достаньте картридж и покачайте его из стороны в сторону, или (если у вас принтер с приводом Canon) вытащите тонер-картридж, встряхните его и установите заново.

Замените тонер-картридж. В системах с раздельным расположением барабана и тонер-картриджа может помочь очистка приемного резервуара.

**4. Как избежать появления черных вертикальных линий или полос.**

Очистите главный коронирующий провод.

Замените единый картридж или тонер-картридж в другом типе принтеров.

Замените барабан.

**5. Не пропечатываются части страницы.**

Проверьте бумагу на загибы.

Проверьте тракт подачи бумаги на предмет отсутствия обрывков бумаги.

**6. Появление на бумаге случайных следов, темных пятен и т. п.**

Проверьте уровень тонера. Посмотрите, не включилась ли сигнальная лампочка.

Очистите тракт подачи бумаги, коронирующие узлы.

Очистите внутренние механизмы принтера.

лишь слегка протирать их чистой тряпкой. Представители фирмы Canon утверждают, что пользователи могут протирать "внутренности" принтера влажной тряпкой или использовать маленький ручной пылесос. Другие производители, напротив, отдают предпочтение сухой тряпке. Ни в коем случае не используйте сжатый воздух! Иначе в принтере окажется пыли больше, чем будет извлечено.

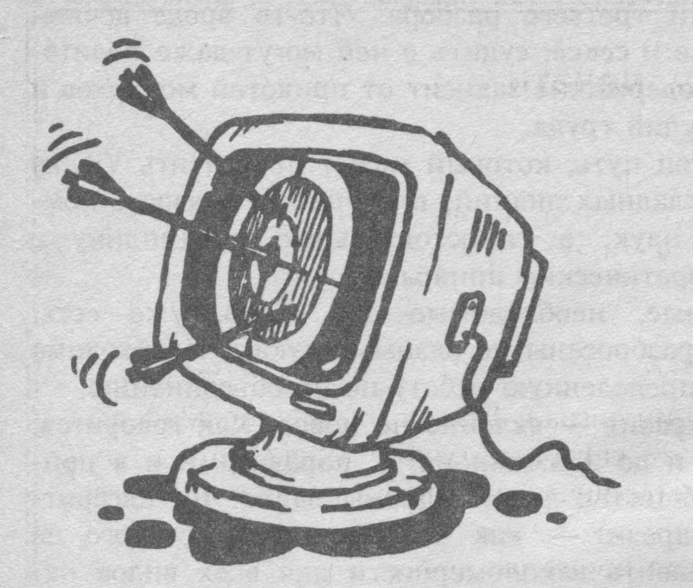
Условия эксплуатации также влияют на работу лазерного принтера. Если принтер установлен в помещении, где не поддерживается постоянный температурно-влажностный режим, а воздух не обеспыливается или повышен уровень вибрации, то он нуждается в более внимательной проверке и частом обслуживании. Бумага низкого качества, легко подвергающаяся всевозможным повреждениям, также учащает необходимость операций по поддержке качественной работы принтера. Не стоит хлопать крышкой принтера, как дверью после ссоры — прецизионная оптика этого не любит.

## Пара слов о смазке

Если ваш принтер начал подозрительно скрипеть, это вовсе не означает, что его нужно срочно смазать. Вовсе нет. Подобные звуки, как правило, пропадают немедленно после замены старого барабана. Однако если ваш принтер изрядно "постарел" или имеет на счету более 150,000 отпечатанных страниц, то он уже нуждается в небольшой подстройке механических узлов. Пригласите специалиста. Сами вы можете смазать не те детали, использовать неподходящее масло или использовать его в слишком большом количестве, что может привести к неприятностям; или повернуть винт, настраивающий совсем не то, что вы думали.

Разумеется, приобретая лазерный принтер, вы надеялись на его высококачественную работу и блестящие возможности. Если вы не будете пренебрегать процедурами по обслуживанию, он не обманет ваших ожиданий.

*И.Вязаничев*



Ну когда же оно кончится...

# Долгое детство компьютерной вирусологии

**ViL** *(virusology)* — компьютерная вирусология.
**Vi** *(virus)* — компьютерный вирус.
**iVi** *(ideal virus)* — идеальный вирус.
**aVi** *(auto virus)* — самомодифицирующийся вирус.

*Уважаемая редакция, не совсем обычная история этого материала требует нескольких предварительных слов.*

*Знакомый системный программист попросил меня написать задиристое, открытое письмо для компьютерной сети, как ответ на непарламентские фигуры речи у законодателей отечественной вирусологии. Предполагалось также, что письмо оживит дискуссии по компьютерным вирусам.*

*С удовольствием взялся за дело. Однако вначале необходимо было разобраться: что такое паразиты вообще, и компьютерные в частности. Результат перед вами. Выводы оказались совершенно неожиданными и противоречащими, на первый взгляд, здравому смыслу.*

*Заказчик письма остался недоволен: выводы показались ему слишком сумасшедшими, размер будущего послания слишком большим.*

*Содержание статьи можно выразить в двух словах: у паразитов всех мастей в природе и обществе появился интеллектуальный собрат — Vi, причем с мощным мат. аппаратом для описания. Возможно, это послужит сигналом для объединения всех аналогичных явлений из разных дисциплин в единую науку, контуры которой я попытался представить здесь.*

*С другой стороны, именно междисциплинарные усилия позволяют объяснить природу и роль Vi, что невозможно сделать, исходя только из компьютерной вирусологии.*

*Выводы работы предполагают смену парадигмы — отход от образа вируса-врага. Реакция моего знакомого характерна для всех программистов-профессионалов, с которыми я обсуждал этот материал: от полного неприятия вначале, до дружеского сочувствия после долгих объяснений.*

*Умеренная темнота отдельных наукообразных мест не должна испугать возможного читателя, а задорный, ироничный тон, как мне кажется, соответствует стилю вашего журнала.*

*И хотя некоторые технические детали несколько устарели — работа над материалом началась год назад — он еще достаточно актуален, иначе я не стал бы его вам предлагать.*

*С уважением, В.Веселов.*

## Вместо предисловия

Перелистываю приложение к очередному AIDSTEST'у — все как и раньше, только реклама стала более навязчивой. Ловлю себя на мысли о том, что вирусология засиделась в пеленках. Если человек может позволить себе топтаться на месте, то для науки это не естественно.

О принципиально новом подходе к компьютерным вирусам — эта статья.

## Современная вирусология

Прежде чем рассматривать новую модель вирусологии, кратко остановимся на том, что имеем на сегодня.

На заре советской компьютеризации антивирусные программы распространялись бескорыстно, в порядке братской помощи. От борьбы с Vi веяло романтикой, казалось, еще одно общее усилие — и с этим видом программ будет покончено. Переход грандов отечественной ViL от коммунистических отношений к

рыночным показал, что борьба будет долгой и прозаической.

Тем не менее Vi не потеряли своей привлекательности, о чем говорит неиссякающий поток литературы. Несмотря на огромный практический интерес и многочисленные публикации, наука ViL никак не может окончательно родиться.

Какого бы автора мы ни взяли — от прагматичного Лозинского до грозно-назидательного Безрукова — они не дают в своих публикациях удовлетворительного ответа на принципиальные вопросы, касающиеся самой сущности Vi и перспектив их развития. Случаются отдельные научно-мистические робинзониады, но они либо еще больше запутывают предмет, либо просто являются переложением заповедей христовых на компьютерную тематику.

Большинство авторов вообще не поднимаются выше классификации, принципы которой еще не выработаны. В зависимости от пристрастий автора за основу классификации принимается длина Vi, способ атаки, эффект проявления и т.д.

Со средствами защиты дело обстоит не намного лучше. Хотя основные виды защиты определились (Детектор, Страж, Ревизор), в литературе нет объективного сравнительного анализа их эффективности. Да и какая может быть объективность, если большинство авторов рекламируют свои собственные защиты. И как следствие, популярны статьи типа: "вы все неправы — покупайте мою защиту" (например, Ф.Н. Шерстюк [БИТ № 2´91] ).

О перспективах развития различных типов защит, даже на ближайшее будущее, и говорить не приходится.

Сейчас стало модным начинать статьи о Vi, как партийную анкету, словами "я вирусы не писал и с их авторами не знаком" (например, Е.Касперский [КомпьютерПресс № 5´91]), а далее следуют подробные рассуждения о Vi и их авторах.

Клятвы на верность и самореклама — признаки незрелости вирусологии, от которой она должна со временем избавиться.

## Два пути вирусологии

Современное состояние ViL не случайно, на это есть уважительная причина. Вирусы — крайние в цепочке "аппаратура — ОС — ПО". Даже самые незначительные модификации в этой последовательности могут радикально изменить классификацию Vi.

Делать какие-либо обобщения на "текучем" вирусном материале невозможно: легкий ветерок аппаратных перемен разрушит до основания (то есть до классификации) карточный домик любых теоретических построений. И все придется начинать сначала, но уже с новыми типами Vi.

Дело здесь не в молодости ViL как науки. Если строго придерживаться стандартного пути развития — от анализа накопленного материала, то ViL ожидает участь науки третьего разбора. Что-то вроде вечной темы о семье и сексе: судить о ней могут даже политэкономы, а содержание зависит от прихотей монархов и развития орудий труда.

Есть другой путь, который может превратить ViL из набора прикладных знаний, плетущихся в хвосте компьютерных наук, в самостоятельную дисциплину с мощным теоретическим аппаратом.

Все данные, необходимые для этого, уже есть, только они разбросаны по разным наукам. Необходимо проделать определенную работу по их объединению.

Vi как паразит — явление не новое. Как говорится, были герои и до Агамемнона. С паразитами и в природе, и в обществе люди знакомы давно. Но алгоритмический паразит — как раз то звено, которого не хватало. Выявить закономерности для всех видов паразитизма (биологического, социального, компьютерного) и спроецировать их на ViL — дело техники.

## Основы паразитизма

Что же может быть общего у бактерии, мафиози и отрывка кода для ЭВМ?

Необходимое условие для всех указанных видов паразитизма — существование в системе по крайней мере двух подчиненных, один другому, уровней управления. При этом важны не сами уровни, а несоответствие между ними, на котором можно паразитировать. Под несоответствием понимается многовариантность реализаций младшим уровнем команд старшего (неполнота, как сказал бы математик). Чем больше такое несоответствие, тем больше экологическая ниша для паразитов.

Действительно, в государстве преступность существует благодаря незамысловатой схеме: до бога высоко, до царя далеко. Многочисленные промежуточные эшелоны власти управляют обществом в целом, но контролировать действия каждого члена очень сложно. Недаром говорят о правосудии Pede Poena claudo — хромоногая, то есть медлительная, кара.

Распространение паразитов через руки (например, при рукопожатии) обусловлено тем, что рука управляется через несколько уровней посредников (нервный, мышечный, кожный, клеточный и т.д.). Каждый следующий уровень управления, выполняя приказы предыдущего, имеет некоторую свободу действий, которая и дает паразитам шанс на существование.

Наверное, реален уровень, ведущий строгий учет и контроль всех закоулочков рельефа ладоней, а также лояльности каждой клетки кожи. В этом случае болезни грязных рук искоренятся, но шевелить руками каждый день будет сложно.

Обычный пользователь управляет ЭВМ тоже через уровни-посредники. Цепочка "код (ассемблер) — ОС — программы — оболочки" облегчает работу пользователя. Однако за удобства приходится расплачиваться: любой из перечисленных уровней может стать нишей для соответствующих типов Vi.

Воистину, сам себе лучший слуга: вместе с исполнителями рискуешь получить и паразита.

Механизмы развития паразитов используют определенные дефекты структуры управления системы-хозяина (неполноту уровней управления). Если система деградирует и ее управление разваливается или, наоборот, она очень торопится в развитии, громоздя уровни управления один на другой, то естественные при этом пустоты в управлении — настоящие экологические ниши для новых систем-паразитов.

Неполнота уровня управления — это что-то вроде развилки дорог. Исполнитель может выбрать любую из них. Если возможностей слишком много, на исполнительном уровне начинаются процессы самоорганизации, то есть появляются паразиты.

Таким образом, все виды паразитической деятельности можно определить как процессы самоорганизации в уровнях управления системы-хозяина.

Не следует понимать это так, будто самоорганизация обязательно начинается с молекулярного уровня. Если появляется возможность существовать за чужой счет, желающие всегда найдутся, и не обязательно это будут молекулы.

Если попытаться ответить на вопрос: паразиты — это добро или зло, то получится удивительная картина. Оценка роли паразитов будет зависеть от того, какая система-хозяин рассматривается.

Действительно, в биологических системах на уровне отдельных особей паразит, как правило, наносит вред организму хозяина. На уровне популяции хозяина гибель части особей оказывается полезной. Действуя по принципу отрицательной обратной связи, паразит регулирует численность популяции, повышая устойчивость системы "паразит-хозяин". Аналогичный механизм работает на уровне биоценоза. Паразиты дискриминируют неустойчивые виды на благо всей экосистемы.

Это позволяет объяснить отсутствие эффективных механизмов защиты от паразитов у большинства видов (а у некоторых видов млекопитающих даже приспособления, облегчающие паразитизм).

В обществе картина повторяется. На уровне отдельно взятого нашего Отечества в период застоя




предприниматели (с официальной точки зрения) — паразиты, мешающие идти к светлому будущему. На уровне же мирового сообщества к бизнесу отношение противоположное. В государстве бывшие разбойники иногда становятся опорой престола, как казаки или корсары. Большинство политических и религиозных движений начинались как незаконные, то есть с точки зрения государства как паразитические.

Понятно, что с точки зрения системы-хозяина, паразит он и есть паразит: отбирает с трудом добытые ресурсы, мешает развитию и т.д. Однако на уровне сообщества систем роль паразитов, как мы знаем, иная.

Если члены сообщества прошли "проверку паразитами" (то есть не имеют значительных дефектов в структуре управления), вся совокупность будет более устойчивой.

Это все равно, что собирать дом из крепких проверенных кирпичей: хотя и дольше, чем из первых попавшихся, зато надежнее. В противном случае быстро возведенное здание может рухнуть при нетривиальном стечении обстоятельств, например, при одновременном закрывании нескольких дверей и одной форточки.

Сообщества, избавленные от пресса паразитов, внешне развиваются быстрее. Но весь эволюционный путь сообщества может быть перечеркнут внезапно начавшимися процессами самоорганизации — социальная или биологическая катастрофа отбросит сообщество назад, и многое придется начинать сначала.

Таким образом, паразитизм является стабилизирующим фактором для систем-сообществ, состоящих из равноправных членов.

С точки зрения стороннего наблюдателя, наличие паразитов в системе показывает ее недостатки и возможные направления развития.

Например, СПИД вскрыл недостатки иммунной системы: несовершенство структуры управления делает ее беззащитной перед собственными паразитами. А характер экономической преступности в годы застоя позволял судить о нашем будущем еще тогда.

Что же приобрела ViL в результате общесистемных рассуждений? Чтобы ответить на этот вопрос, необходимо сначала определить сам предмет разговора — Vi.

## Определение Vi

С определением Vi в современной ViL сложилась забавная ситуация. С одной стороны, ViL, как любая наука, не может обойтись без определения объекта исследования, с другой — накопленного материала не достаточно для конструктивного определения. В этих условиях исследователю приходится полагаться только на собственное чутье, что выражается большим количеством различных определений, встречающихся в литературе.

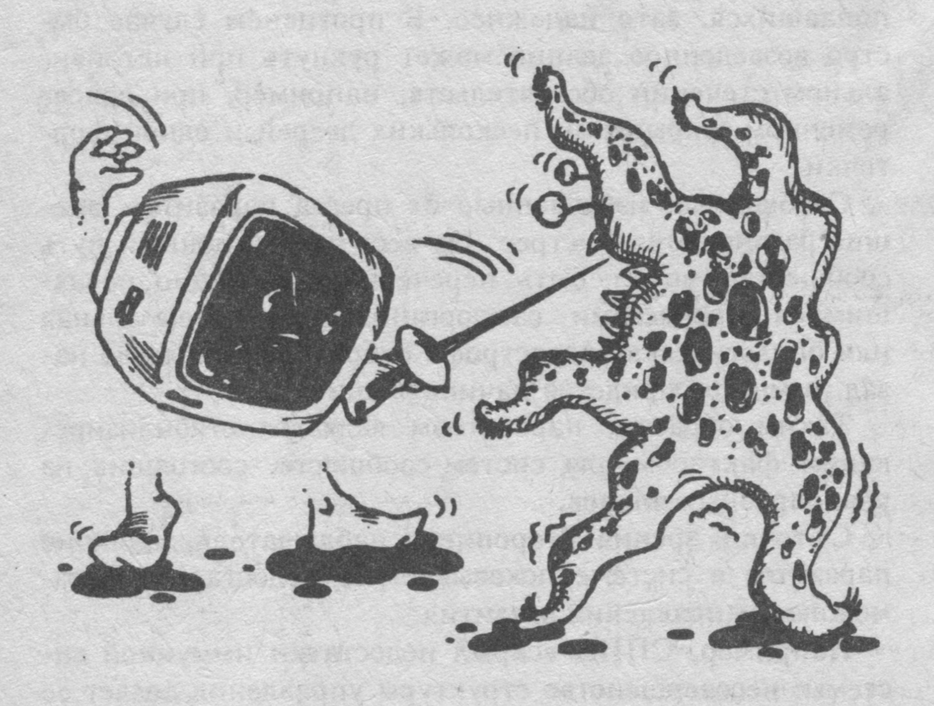
Иногда за основу определения просто берется одно из свойств Vi, которое представляется автору наиболее значимым: скрытность (И.Моисеенков), вандализм

(Н. Безруков), возможность внедряться в программы (Ф.Коуэн) и т.д.

Некоторая преемственность все же наблюдается. Большинство определений в отечественных публикациях являются просто перепевами определения Коуэна. Из публикации в публикацию в различных модификациях кочует маленький филологический шедевр западного исследователя: "...копии, не обязательно совпадающие с оригиналом..." (Е.Касперский). Доверчивость наших авторов могла бы внушать уважение, если бы не внушала тревогу.

В современной литературе определение Vi пока выполняет функции декоративного украшения статей, а не инструмента исследований, что отражает общее состояние ViL.

Формулировка определения потребует некоторых усилий, которые, впрочем, быстро окупятся.



Рассмотрим систему, состоящую из компьютеров и пользователей, а также множество алгоритмов (А) для данной системы, обладающих следующим свойством: возможным результатом работы любого алгоритма из А является создание алгоритма, принадлежащего А. Элемент из А будем называть идеальным Vi (iVi — ideal Vi).

Может показаться, что такое определение — тавтология (Vi есть Vi), но это не так. Математики называют подобные конструкции рекуррентными определениями. Достаточно только указать, что данное множество алгоритмов не пусто. Но об этом, наверное, знает каждый читатель. (Например, Vi, которые не модифицируют своего кода при работе, то есть порождают самих себя.)

Кроме iVi, которые могут порождать себе подобных до бесконечности, существуют Vi, размножающиеся до выполнения определенного условия (число делений, дата и т.д.), при котором возможно самоуничтожение, (вместе с содержимым диска), и следовательно, формально не удовлетворяющие определению iVi.

Такие Vi получаются, как правило, путем добавления к iVi условия и алгоритма проявления. Определение, обобщающее свойство iVi и Vi с размножением по условию, включает в себя понятие совокупного числа потомков, а также разумного критерия для оценки опасности такого числа в конкретных условиях. Так как это определение слишком громоздко и не добавляет ничего интересного по сравнению с определением iVi, оно здесь не приводится. К тому же именно алгоритм проявления Vi (в отличие от алгоритма размножения) сильно зависит от психологии его создателя (чувства юмора, степени извращенности, глубины обиды...), и поэтому интереса не представляет.

Таким образом, говоря по-простому, *Vi — это алгоритм для системы ЧЕЛОВЕК + ЭВМ, способный порождать себе подобных достаточное число раз.*

Любопытно, что понятию Vi в приведенном выше смысле соответствует сумасшедший оператор, который тайком распространяет (копирует) какую-либо программу, при этом основная часть алгоритма такого Vi находится в голове оператора. Как ни парадоксален этот пример, известный Vi "Новогодняя елка" работает очень похожим образом: оператор САМ, по вежливой просьбе Vi, запускает его на исполнение.

Здесь уместно сделать несколько замечаний.

Нельзя ставить знак равенства между компьютерным паразитом вообще и Vi в смысле данного определения. Возможность размножаться не главное для паразита, так как есть и другие пути самоорганизации, например, программа, которая в результате ошибки постоянно увеличивает свою длину, то есть строит свои собственные независимые структуры. Аналог — раковая опухоль, которая другим организмам не передается, но для организма-хозяина является паразитом.

С другой стороны, возможно применение вирусных программ для оптимизации работы сетей, защиты от Vi и других полезных целей, что не совместимо с понятием паразитизма.

Тем не менее на сегодня предложенная формулировка хорошо описывает известные виды компьютерных паразитов, и тонкостями можно пренебречь.

Главная особенность данного определения, в отличие от остальных, переход к системе ЧЕЛОВЕК + ЭВМ. Это не догматическое исправление, навеянное присутствием человека возле машины. Как будет показано далее, противоречие между двумя компонентами этой системы и породило Vi. Противоречие между декларативным языком человека и процедурным языком машины.

## Причины возникновения и перспективы развития Vi

Официально признанная точка зрения на природу вируса проста и тенденциозна: с одной стороны, существование Vi предсказано и обосновано Н.Виннером, а спровоцировано размахом компьютеризации, с другой стороны — все существующие Vi созданы конкретными

авторами (Иванов, Петров, Моррисон), и если бы этим программистам привили в детстве чуть больше любви к ближнему, то и не пришлось бы создавать хитроумные защиты. После иных публикаций может даже сложиться впечатление, что несколько удачных педагогических приемов при поддержке строгих уголовных законов — и с Vi (не сразу, но в принципе) будет покончено. Однако, если посмотреть шире официальной точки зрения, то к радужной картине (солнце, пальмы, золотой песок) придется добавить чуть-чуть зеленых крокодилов.

Вся прелесть морализации в ее доступности. Типичная поучительная сентенция "делать Vi — все равно, что мочиться в лифте" — то ли о воспитании молодежи, то ли о неустроенности быта. Кажется, если покрепче обозвать — и объяснять нечего. Недостаток подхода — полное непонимание того, что происходит, и ограниченный диапазон действий (можно лишь ругаться и писать защиты от все новых и новых Vi).

Как это часто бывает, если отделить науку от эмоций, открываются механизмы явлений, мало связанные с культурой и воспитанием.

Язык ЭВМ (машинный код или ассемблер) оказался настолько неудобен для человека, что для нормального управления ЭВМ пришлось создать несколько промежуточных уровней: ОС, языки высокого уровня, оболочки.

Конечно, машинный код не специально разрабатывался таким тарабарским по прихоти чудаков-инженеров. Когда выяснились возможности машины Тьюринга, предтечи современных ЭВМ, никто и не думал о будущих трудностях общения человека и машины.

В современных персоналках воображаемая лента (машины Тьюринга) достигает длины сотен километров (с отверстиями через 1 см) и движется со скоростью многих километров в секунду. Понятно, что человек не может без посредников управлять такими процессами. А раз появляются посредники, то необходимое условие для существования паразитов соблюдено.

Как показывает опыт, минимальная длина самораспространяющегося алгоритма составляет 100-1000 байт в зависимости от типа ЭВМ (не для сетей). Этого достаточно для существования Vi, так как длина большинства прикладных программ на 2-3 порядка больше, и затеряться в таком объеме не сложно. (Полная формулировка достаточного условия выходит далеко за рамки статьи. Здесь придется ограничиться небольшим замечанием о соотношении длин.)

В системе ЧЕЛОВЕК + ЭВМ изначально существовал огромный разрыв между языком человеческого общения и языком ЭВМ, порожденный архитектурой Неймана-Тьюринга. В процессе развития разрыв постепенно заполнялся промежуточными уровнями, которые образовали многоярусную экологическую нишу; при этом каждый уровень образует нишу для соответствующего типа Vi.

Очевидно, что максимальные возможности для Vi предоставляет самый нижний уровень управления ЭВМ (машинный код). Вирусы, написанные на ассемблере, составляют подавляющее большинство известных Vi. При этом для системы, состоящей из нескольких подсистем ЧЕЛОВЕК + ЭВМ (например, компьютерная сеть), язык самого низкого уровня, на котором еще может быть написан Vi, должен удовлетворять добавочному условию — быть одинаковым для всех машин сети.

Таким образом, авторы Vi выполняют лишь повивальные (как сказал бы классик) функции, то есть только помогают Vi появиться на свет, окрашивая в меру своих способностей этот процесс в субъективные тона. Основная же причина появления Vi — принципиальные отличия языков человека и ЭВМ.

Здесь уместно следующее. На кухне даже самому большому таракану не дадут управлять всем хозяйством ни секунды. Вирус же получает в свое распоряжение все ресурсы ЭВМ, что делает его исключительно опасным. Тенденции распараллеливания работы машины и организация привилегированного доступа в память превратят Vi из разбойников в нахлебников.

Разработка новых технологий общения человека и ЭВМ (экспертные системы, искусственный интеллект) направлена на создание языка высокого уровня, близкого к языку человеческого общения, и вытеснение языков более низкого уровня, а значит на подрыв самих основ существования Vi. Из сказанного следует, что через несколько десятилетий ViL как дисциплина о Vi низкого уровня (а возможно, и об алгоритмических Vi вообще) перестанет быть актуальной.

Может быть, кому-то станет жалко Vi. Ничего не поделаешь — обмен алгоритмами (в отличие от обмена данными) на нескольких уровнях в будущем скорее всего не понадобится. Достаточно будет сформулировать задачу на обычном (человеческом) языке, а ЭВМ сама создаст максимально эффективный алгоритм.

Однако свято место пусто не бывает. Скоро (по историческим масштабам) может образоваться ниша для электронно-алгоритмических паразитов: машины рано или поздно станут обмениваться не только информацией, но и “железками“, тут-то их и подстережет извечный спутник развития — паразит.

## Vi — это хорошо или плохо?

Как указывалось ранее, оценка роли паразита зависит от выбора системы координат. С точки зрения отдельной ЭВМ, вывод о вредоносности Vi напрашивается сам собой. Не нужно много усилий, чтобы превратить его в абсолютную истину.

Стараниями лихих вирусологов идея о злом Vi, который портит полезные файлы, легко проникла в сердца и кошельки большинства пользователей.

Однако существуют и стратегические проблемы, связанные с развитием информационной индустрии как единого организма.

Vi имеют непосредственное отношение к глобальной устойчивости сложных систем, состоящих из людей и машин. Без решения вопросов, которые вскрыли Vi, невозможны устойчивая работа и нормальное развитие таких систем. Если бы эти вопросы не были поставлены авторами Vi сейчас, они бы возникли без помощи человека позже, например на этапе автоматической разработки программ.

При объединении отдельных машин в большие сообщества роль Vi не так очевидна и требует дальнейших исследований. Но вместо того, чтобы лишний раз проверить: а так ли страшен Vi, как его рисуют западные коллеги, мы занялись любимым российским делом — воспитанием.

За рубежом рано или поздно будут по достоинству оценены созидательные возможности Vi. Со временем к этому мнению присоединятся и наши исследователи. Но будет неплохо, если в нашей полудикой компьютерной стране мы сможем проявить самостоятельную мудрость и терпимость, а не действовать по принципу: все, что Запад не любит, — мы ненавидим.

*В.Веселов*

*(Окончание следует)*

# Математические аспекты компьютерных вирусов

## Соотношение сложности защиты и вируса

Каждый, кто "забирался" достаточно глубоко в операционную систему или большой прикладной пакет, обязательно испытывает удивление — как же все-таки хрупко программное обеспечение! Достаточно изменить команду, маску, байт, бит — и функционирование программы радикально изменяется. Трудно только найти такое место, зато после этого беззащитность программного обеспечения поражает. А если эти изменения носят случайный, неосмысленный характер, реакция программы становится непредсказуемой, но практически со 100% вероятностью программа окажется разрушительной. И для операционной среды, и для себя. С той же скоростью и эффективностью, с какой проходила созидательная деятельность, начнется деструктивная (пожалуй, даже еще быстрее)...

Можно построить очень хитрую систему защиты, но если злоумышленник доберется до той команды, которая передает управление этой системе или проверяет условия вызова этой системы в оперативной ли памяти или в загрузочном модуле на носителе и изменит ее так, чтобы защита всегда обходилась, с чем мы останемся? Сошлемся еще раз на работу Ф.Коуэна, в которой, после обсуждения самых разных алгоритмов блокирования распространения вирусов и защиты программ, Коуэн в разделе "Практические ограничения указанной техники" грустно замечает: "...Другой важной проблемой в ... системах является то, что атакующий может симулировать всю систему, возвращая те результаты, которые и должны быть получены, на самом деле запоминая другую информацию...". И далее: "Другой вариант атаки, который можно предположить, — использование программ, которые симулируют защиту, работая правильно во всех ситуациях, кроме обнаружения распространения инфекции".

Рискнем утверждать, что возможно удастся математически доказать интуитивно очевидный для практических программистов факт, что сложность программы, разрушающей систему защиты, в каком-то смысле всегда меньше сложности системы защиты. С точки зрения философа это очевидно: для работы любой программы нужно поддерживать строгий порядок в данных, если же либо данные, либо код изменить произвольно (по существу, хоть как-нибудь), то произойдет разрушение нормальной работы программы. Поэтому неверно, что несанкционированное проникновение червяка Морриса в систему является только результатом отдельных недочетов в организации разделения доступа. Очевидно, любые системы защиты находятся в невыгодном положении по отношению к вирусам: теоретически перевес снова оказывается на стороне вирусов.

Большие надежды пользователи UNIX возлагают на парольную защиту. Но оказалось, что самым слабым элементом в системах парольной защиты является человек. При использовании самых изощренных алгоритмов кодирования паролей защита вскрывалась относительно просто и эффективно с учетом особенностей человеческой психики, что и продемонстрировал Моррис и чего не учитывали разработчики систем парольной защиты. Способы совершенствования защиты с учетом человеческих качеств очень активно обсуждаются пользователями UNIX, и некоторые предложения уже реализованы в более современных версиях системы. Сложности, связанные с человеком, понятны: легко запоминаются общеупотребительные, банальные слова, таких слов немного и их легко отгадать; а трудно угадываемые случайные (бредовые) комбинации цифр и литер придумываются человеком с большими сложностями и еще хуже запоминаются. Опять-таки человеческое свойство использовать уже использованные пароли (легче запомнить) сильно снижает надежность парольной защиты. Для проникновения в систему червяк использовал словарь всего лишь из 160 английских слов и некоторые простейшие алгоритмы изменения этих слов (типа изменения порядка букв на обратный, изменения больших букв на малые и наоборот). Для того чтобы избавиться от дефектов парольной защиты, свойственных человеку, предлагались и предлагаются различные усовершенствования способа раздачи паролей. Один из наиболее продвинутых в этом смысле проектов осуществлен в MIT (проект CERBERUS, включающий специальный сервер паролей, позволяющий, по замыслу создателей, ликвидировать недостатки схем парольной защиты, связанные

Окончание. Начало в КомпьютерПресс № 10´92.

с наличием такого "неприятного" компонента, как человек).

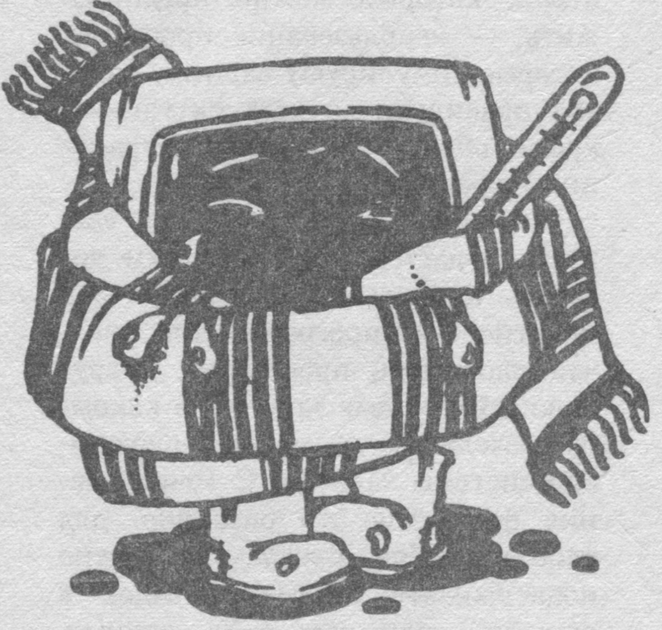
Меры по усовершенствованию парольной защиты предлагаются и для персональных компьютеров. Известны и другие интересные программы идентификации пользователей, основанные на анализе манеры работы пользователя на клавиатуре; утверждается, что эти программы могут не только уверенно распознавать различных пользователей, но и почувствовать стрессовые состояния конкретного пользователя.

При использовании любой системы защиты ключевой вопрос — гарантия реальной работы этой системы защиты, то есть гарантия того, что код самой системы защиты и условия ее вызова остались неизменными. Конечно, современные вычислительные комплексы и операционные системы отошли от "чисто" фон неймановской архитектуры. Особенно многопользовательские многозадачные ОС, которые пользуются аппаратной защитой кода ядра от изменений, делаемых не ядром; есть и более развитые системы аппаратной защиты. Но все-таки, как показывает печальный опыт, этого совершенно недостаточно.

Опять-таки, поскольку вирусы "появляются" в среде самых разных прикладных пакетов в самых разных ОС, считать только ОС ареной борьбы с вирусами — нелепо, необходимо задумываться о проблемах разграничения доступа, защиты информации, целостности кода, защиты от вирусов в среде каждого прикладного пакета.

В настоящее время ситуация осложняется тем, что на отдельных серверах сети работают распределенные базы данных или специализированное обеспечение для печати документов. Последние сами по себе являются развитыми программными продуктами с собственными языками и сетевым сервисом, поэтому они становятся потенциальными заповедниками макровирусов. Но в то же время они тесно связаны с остальными компонентами сети, от которых, очевидно, нужно особенным образом защищать данные. Нам неизвестно, какой сервис реализован на сервере паролей в проекте CERBERUS, но если там есть возможность написания макросов, есть команды копирования — очевидно, в этой среде возможно появление вирусов... Вообще, сетевые вирусы или вирусы в многопроцессорных и особенно распределенных многопроцессорных системах (сообщения о таких системах уже появились) — особенно сложная разновидность вирусов.

Еще червяк Морриса намекнул на возможности в этой области. Насколько можно судить по разбору "дела Морриса" в июньском



номере Communications of the ACM за 1989 год, существовали различные пути распространения инфекции по сети и сам червяк был отчасти распределен в сети (не все заражаемые компьютеры получали полный набор компонентов червяка). Наверное, при должном усовершенствовании можно построить по-настоящему распределенный червяк так, чтобы ни в одном из узлов сети не существовало полного набора компонентов. Понятно, что это чрезвычайно осложнит работу исследователей по выделению и нейтрализации червяка.

Еще больше возможностей в среде многопроцессорных распределенных систем, где только нарушение синхронизации работы процессоров может привести к самым тяжелым последствиям. Эти рассуждения, к счастью, носят пока абстрактный характер (кажется).

Главная реальная трудность сегодня состоит в том (и с этим согласны практически все), что вирусы "рождаются" и "существуют" в среде самых разных программных продуктов, стоит только продукту достаточно усложниться. Обилие мест появления вирусов создает очень большие сложности для защиты: в среде каждого пакета есть свои особенности, которые используют вирусы, "живущие" в среде этого пакета, и необходимы специальные знания о каждом пакете или гарантия неизменности кода пакета, чтобы быть хоть в чем-то уверенным. Одно количество программных продуктов, в среде которых можно писать вирусы, создает ощущение безнадежности. А ведь мы обсуждаем пока только текущее состояние вредоносного программного обеспечения.

И наконец, последний аккорд в этой грустной теме. Вернемся на секунду к обсуждению человеческого аспекта вирусного феномена.

Как мы видели, написать вирус очень просто, вирус оказывается относительно несложной и небольшой по объему программой (если, конечно, не ставить каких-то специальных целей), результаты работы вируса можно наблюдать очень быстро. Этот комплекс свойств делает вирусные программы практически идеальным учебным материалом. Первое, что приходит в голову студенту, только-только научившемуся программировать, — написать свой вирус: это одно из лучших осмысленных упражнений в программировании. В меру сложное, небольшое по объему, быстро демонстрирующее результат.

Увы, это не только наш прогноз. Многие активные борцы с вирусами, выступавшие на Киевской конференции по защите от вирусов, буквально стонали, что их за-

хлестывает поток очень простых студенческих вирусов.

## Почему вирусы появились именно сейчас?

Нам придется представить хоть какие-то объяснения по поводу того, почему именно сейчас разразились пандемии компьютерных вирусов. Ведь какая-нибудь система MVT на больших компьютерах IBM была нисколько не проще DOS! А оборудование, функционировавшее на больших машинах IBM 60—70-х годов, едва ли уступало по сложности современным персональным компьютерам.

Почему же в те, теперь уже давние, безвирусные времена, не было компьютерных эпидемий?

Скорее всего, они были бы, если бы столь сильно, как теперь, были развиты системы межкомпьютерного обмена и связанный с этим сервис.

Действительно, лет 20-25 и даже 10-15 назад межкомпьютерная связь была еще чем-то экзотическим, необыкновенным; отсутствовали стандарты и компоненты операционных систем, оборудование для обмена не было столь совершенным, умным, интеллектуальным. Поэтому речь об инфекциях в компьютерных сетях и не заходила.

Но, может быть, происходили неприятности на отдельных компьютерах?

Несомненно, происходили! Почти в каждом крупном вычислительном центре есть свой апокриф, свои легенды и мифы, свои боги, герои и титаны (как "вскрытия" компьютеров, так и защиты их). Правда, все неприятности были локализованы и не принимали угрожающих размеров, не носили характера эпидемий из-за указанных выше причин. Ну а выносить сор из дома далеко не все любят. Более того, некоторые хорошо известные разработки начинались, как попытки пользователей что-то доработать в ОС, то есть как пиратские проникновения в систему. Например, система HASP начиналась как полулегальное вторжение в OS/360 с целью усовершенствования подсистемы ввода/вывода. Причины, по которым достаточная уже тогда сложность оборудования и программного обеспечения не вызвали к жизни компьютерные инфекции, следует искать в особенностях носителей и ОС.

**1.** Вспомним, что самым популярным носителем еще 7-10 лет назад в нашей стране (15-20 — во всем мире) были перфокарты, чуть позже — магнитные ленты. Вы можете себе представить компьютерный вирус, распространяющийся на перфокартах? В конце концов нетрудно обнаружить странные перфокарты и изъять их из колоды — и у вируса (точнее у его авторов) нет никакой гарантии его распространения.

С магнитной лентой ситуация чуть сложнее. Естественно, на нее можно записать все, но информация на ленте организована последовательно, скрыть ее невозможно. Но самое главное — работа с магнитной лентой и перфокартами осуществляется операционной системой не автоматически, а только по запросу пользователя. И в этой ситуации трудно придумать "гарантированный" способ распространения инфекции.

**2.** ОС недавнего прошлого были ориентированы на пакетную обработку и средства общения процессов, командные интерпретаторы (да и сами программы) писались в расчете на эту идеологию. Такое устройство операционной среды как бы делило жизненное пространство программ на клеточки, практически не позволяя им общаться; ОС не передавала процедурам никакой дополнительной информации (вроде имени вызываемой процедуры, что было связано и с устройством файловой системы). Язык управления был тривиальным, такой важный языковой аспект, как рекурсия, был сознательно подавлен. Стоило только пойти навстречу пожеланиям пользователей (а не пойти было нельзя, поскольку такая простота языков управления заданиями мало кого из пользователей устраивала и они делали многочисленные попытки их развития), которые на больших машинах IBM привели к созданию подсистем JES2, JES3, NJE, устроить совместную работу компьютеров, чуть-чуть усовершенствовать язык управления — и критический барьер был преодолен. И появились вирусы.

Например, доступ к собственному тексту технически сильно упрощает задачу репликации этого текста, но эта возможность очень удобна при программировании и является стандартной в современных ОС, что позволяет практически без затруднений писать вирусные командные файлы на языке управления заданиями!

**3.** Отсутствие или слабое (логически нестройное) развитие файловой системы в пакетных ОС с их расчленением на физические тома, с отсутствием динамического присоединения физических томов как логических частей файловой системы без участия языков управления заданиями создавали чрезвычайно дискомфортную среду для распространения вирусных инфекций: приходилось увязывать слишком большое количество компонентов ОС для обеспечения собственного размножения при переходе с тома на том, а это



немедленно вызывало рост размера и сложности вирусного кода. С совершенствованием концепции файловых систем это согласование и увязка стали нормальными функциями ОС, и инфекции ничего не нужно было "носить с собой" — все уже находилось в компьютере и оставалось только воспользоваться имеющимся сервисом в своих целях.

## Особенности IBM-совместимых компьютеров и вирусный феномен

Хотелось бы особенно отметить роль фирмы IBM и ее персонального компьютера PC, который ярко и выпукло (из-за особенностей конструкции), но необыкновенно дешево демонстрирует опасности вирусного феномена.

Заметим, что все многочисленные вирусные истории, скандалы и происшествия в популяции IBM-совместимых компьютеров не вызвали такого резонанса в среде программистов, правительственных чиновников и просто обывателей, как один-единственный паралич сети, случившийся из-за сравнительно безобидного "червяка Морриса". Если вирусы в популяции IBM-совместимых компьютеров могут стирать файлы, портить файловую систему, уничтожать информацию, то "червяк Морриса" ничего такого не делал, а только размножался, отбирая ресурсы сети, пока они еще оставались. Финансовый ущерб, вероятно, прямо пропорционален поднятому шуму.

Упомянутое отличие популяции IBM-совместимых компьютеров состоит в широком, продуманном, хорошо спланированном использовании базовой системы ввода/вывода (BIOS), которая находится в ПЗУ. Причем программы BIOS´а постоянно совершенствуются, возможности их все время расширяются. Такие программы размещают в самых разных местах: кроме "главного" BIOS´а, используют BIOS´ы различного стандартного (графических адаптеров, контроллеров дисков) и нестандартного оборудования.

Это наиболее быстрый, надежный и наименее обременительный или заметный для пользователя способ изменения конфигурации системы. Но, с вирусной точки зрения, постоянно находящийся в компьютере, готовый к употреблению BIOS, особенно та его часть, которая обслуживает дисковый ввод/вывод, подобна снаряженной бомбе, заложенной в компьютер.

Здесь уместно вспомнить еще одну старую (1936 г.) работу математика Геделя, в которой рассматривается проблема длины доказательств (или, в нашей терминологии, программ).

В определенных условиях верны следующие факты:

а) существуют выводимые формулы (результаты), вывод которых (программа для получения результата) сколь угодно более сложен, нежели сама формула;

б) если добавить к аксиомам любую независимую формулу (программу для получения результата), то найдутся выводимые формулы (результаты), вывод (программа для получения результата) которых при прибавлении независимой формулы будет сколь угодно проще, чем без новой формулы.

Наличие BIOS можно рассматривать, как расширение исходного набора формул (машинного языка) независимыми формулами (программами BIOS´а), а это приводит к резкому упрощению программ ОС, связанных с (дисковым) вводом/выводом. Это приятно, но в той же мере упрощаются и программы вредоносные, а это уже очень опасно. Поскольку спрятать маленькую программу или написать программу, которая при некоторых условиях строит маленькую вредоносную программу, оказывается очень и очень несложным делом, причем использование стандартов BIOS´а автоматически (BIOS для этого и был задуман!) сделает эту вредоносную программу переносимой с компьютера на компьютер и независимой от конкретного оборудования.



Вероятно, тот же результат Геделя может объяснить, почему наименьшие размеры имеют вирусы, написанные на языке командных файлов или макросов прикладных пакетов, — используется расширение машинного языка независимыми (?) формулами — программами прикладного пакета.

## Маскировка вирусов

Последний вопрос, который хотелось бы затронуть до того, как подводить итоги, — вопрос о маскировке вирусов. Неутешительные теоретические результаты получены в работе Л.Адлемана даже для простых случаев. Подробный практический анализ случаев, когда весь вирусный код предварительно построен, приведет нас к печальному выводу, что возможности "тривиальной" маскировки настолько велики, что задача обнаружения вирусного кода в программе в общем случае едва ли разрешима.

В самом деле, пусть вирус работает на уровне системных запросов — то есть ему необходимо выдать несколько вполне определенных последовательностей машинных команд. Но между любой парой действительно нужных ма-

шинных команд всегда можно вставить произвольное количество пустых команд (NOP — нет операции) или ничего не меняющих пересылок из регистра в регистр, или еще какие-то произвольные фрагменты кода, ничего не меняющие в состоянии процессора. Очевидно, что даже в такой сравнительно несложной ситуации для корректного обнаружения вирусного кода детектор должен обладать возможностями дизассемблирования с элементами семантического разбора или псевдоисполнения. Только на таком уровне можно корректно идентифицировать программы, выдающие потенциально опасные (вирусные) запросы системе. Эти трудности понимают конструкторы детекторов, во всяком случае в уже многократно цитированном здесь журнале Computers & Security мелькнуло сообщение о программе VIRSCAN, которая выдает предупреждение о вирусной инфекции, если 75% кода (?) совпало с маской, по которой происходит поиск. Несколько исследователей в СНГ пытаются как-то реализовать "универсальный" детектор вирусов, который содержит встроенный дизассемблер и пытается идентифицировать потенциально опасные программы не по каким-то кодам, а по состоянию псевдорегистров во время псевдоисполнения программы (понятно, о каких накладных расходах и о каком качестве детектирования идет речь).

К сожалению, и в этом вопросе авторы вирусов опередили конструкторов антивирусов. Существуют зашифрованные вирусы, которые при дизассемблировании и псевдоисполнении работают со своим кодом как с данными, и в этом случае формальные приемы мало помогают, нужно участие специалиста.

Вообще возможности маскировки вирусного кода даже в локальной системе — очень интересный и мало исследованный вопрос. Можно сослаться на результаты, полученные В.Голодным на примере популяции IBM-совместимых компьютеров под управлением DOS. Он обнаружил несколько красивых нетривиальных механизмов маскировки, распространения и хранения вирусного кода. Часть механизмов маскировки использует знание особенностей конкретных поражаемых программ — это не укладывается в рамки абстрактных моделей компьютерных вирусов, рассмотренных Ф.Коуэном и Л.Адлеманом.

Хотя исследования только начаты, результаты демонстрируют, сколь трудной и неблагодарной может оказаться задача поиска вирусного кода даже в частном случае операционной системы DOS.

В сообщении Бончева на Киевской конференции по защите от вирусов в MS-DOS прозвучала информация о болгарском вирусе PHENIX, который использует столь изощренный алгоритм заражения и своего шифрования, что практически не имеет инвариантных байтов в своем теле.

И наконец, последнее замечание по поводу маскировки. Летом 1991 года практически одновременно в нескольких городах появился вирус, получивший название dir. Этот вирус реализует необычный способ заражения файлов (подробности об этом вирусе см. КомпьютерПресс № 9'91). В каком-то смысле он является гибридом файловых вирусов, поражающих исполняемые файлы, и бутовых, поражающих файловую систему в целом. Вирус заражает исполняемые файлы, не меняя в заражаемом файле ни байта, и в то же время на всю файловую систему имеется только один экземпляр вируса. Вирус не "работает" с исполняемыми файлами, а модифицирует файловую систему, и, наверное, поэтому даже универсальные антивирусные программы не считали его вирусом (опасных кодов модификации исполняемых файлов вирус не содержал!). Схема заражения, используемая в этом вирусе, хотя и красива, но не совсем корректна с точки зрения маскировки в том смысле, что при просмотре на незараженной машине пораженная файловая система оказывается дефектной; заражение же новой файловой системы при малейшей неаккуратности происходит так быстро, что недостаточно опытный пользователь не успевает осознать произошедшее. А на зараженной машине файловая система выглядит нормально, и создается иллюзия случайного сбоя компьютера. Попытки разобраться в ситуации с привлечением новых компьютеров только способствуют распространению инфекции. Насколько мы можем судить, универсальные обнаружители новых вирусных инфекций в этой ситуации проявили себя плохо, а люди справились с этой творческой задачей отлично.

## Некоторые итоги

**1.** Свойства саморазмножения и самоуничтожения являются естественными и неотъемлемыми свойствами любого достаточно сложного конечного автомата. Уровень сложности, с которого эти свойства начинают проявляться, — почти тривиален. Современная вычислительная техника давно превзошла этот уровень, и многочисленные вирусные проявления хорошо иллюстрируют приведенные выше утверждения.

Проблема состоит даже не столько в противодействии вирусному феномену, сколько в сохранении работоспособности больших программно-аппаратных систем, которые, как и большие физические конструкции (аналогия прямая), не выдерживают собственной тяжести и расползаются.

**2.** Наращивание сервиса, совершенствование языков, встроенных систем BIOS, чем озабочены буквально все разработчики, в первую очередь и автоматически приводят к росту вирусной опасности.

Вирусные и троянские программы становятся все более и более компактными и универсальными, все более и более изощренными, причем не столько из-за происков хакеров, сколько по "вине" разработчиков более совершенного сервиса, которые со-

здают для хакеров все более совершенный, богатый и универсальный инструментарий для распространения вирусов и деструкции систем; особенно это относится к распределенным системам с развитым распределенным сервисом (X-Windows), позволяющим создавать распределенные вирусы.

**3.** Рост совершенства и сложности прикладных пакетов, появление развитых языков для таких прикладных пакетов создают отличные возможности для конструирования, написания и появления множества специализированных вирусов и троянских программ, действующих и распространяющихся только в среде этого прикладного пакета и привязанных в основном к этому пакету, а не к конкретному оборудованию. Просматривается возможность "комплексных" вирусов, начинающих работать в каком-то прикладном пакете и "кочующих" в другой пакет или на системный уровень.

Благородная, но, как мы уже выяснили, необыкновенно трудная задача борцов с вирусами многократно усложняется расщеплением на множество задач, требующих дополнительных специальных знаний (то есть своих узких специалистов, своих приемов, своих программ и т.д.).

**4.** Конечно, компьютерные вирусы создаются людьми, и в этом смысле человеческий фактор имеет абсолютный приоритет. Но есть принципиальная разница между двумя ситуациями:

а) когда написать вирус — трудно, для этого нужны обширные специальные знания и высочайшая квалификация и

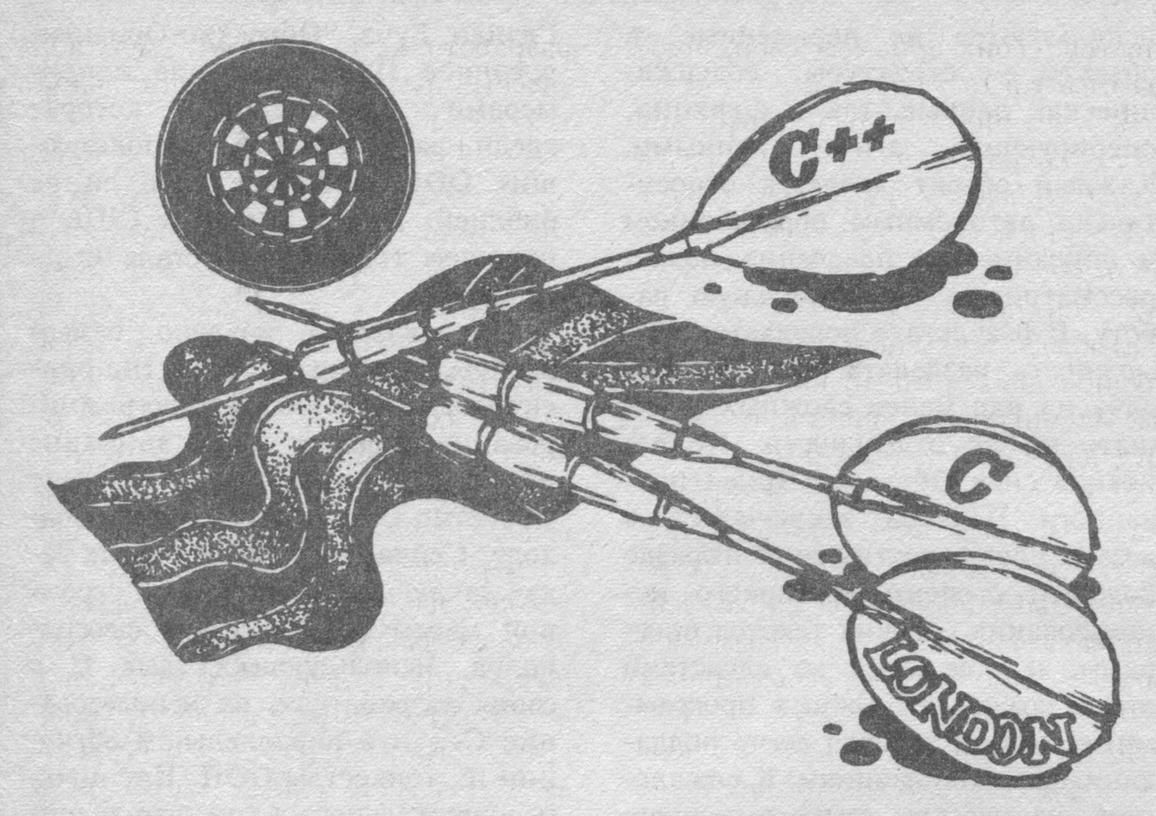
б) когда любой пользователь, знающий язык Shell (Troff, Lotus, dBase, awk и т.д.), может быстро сознательно соорудить вирус или написать вирусную программу по ошибке (см. пример командных файлов в статье).

**5.** Еще одна важная особенность феномена компьютерных вирусов состоит в том, что деятельность создателей вирусов и антивирусных средств совершенно не продуктивна (не приводит к созданию чего-то нового или к более глубокому пониманию природы). Эта деятельность — всего лишь рекурсивные игры с самим собой, и более всего она напоминает игру в шахматы или в карты без партнера. В то же время это смертельно опасное занятие (как мы видели, природа покровительствует разработчикам вирусов), могущее завести развитие вычислительной техники в тупик.

После подведения грустных итогов возникает сакраментальный вопрос: "Что же делать?" Изложенное выше не допускает наивной мысли о том, что все дело в недостаточном количестве квалифицированных программистов, занимающихся антивирусными проблемами, — вопрос значительно более серьезен: нужны теоретические исследования возможности создания устойчивых систем; нужны комплексные разработки — и аппаратные, и программные; нужно изменение стратегии в разработке программного обеспечения с учетом свойств саморазмножения и самодеструкции; нужны радикальные меры, поскольку те идеи, которые сейчас преобладают и реализуются, почти не дают эффекта, напротив, легко просматриваются тупики, в которые они заводят программистскую общественность и проблему.

Поистине эти идеи носятся в воздухе.

*И.Карасик*




# Один в поле не воин

Конференция "C plus C++ in Action", на которой автор этой статьи был единственным, к сожалению, представителем стран СНГ, прошла в Лондоне с 8 по 12 июня сего года. Эта конференция являлась, по сути, первой, посвященной исключительно языкам C и C++, хотя вопросы, связанные с развитием C++, активно обсуждались и ранее, обычно в рамках конференций серии SCOOP. Встреча была организована Бостонским университетом с привлечением в качестве спонсоров ведущего английского журнала для программистов Program Now и фирмы Borland и относилась к разряду событий, о которых в нашей компьютерной прессе вы почти не найдете упоминаний. Этот факт объясняется, в частности, тем, что подобные конференции, проводящиеся с завидной регулярностью в США и в странах Западной Европы, предназначены в основном не для бизнесменов, а для разработчиков программного обеспечения, то есть, пользуясь нашей терминологией, для обычных программистов. По известным причинам, практически никто из российских представителей этой "древнейшей" профессии пока не в состоянии спланировать ни свой собственный бюджет, ни бюджет своей организации с учетом стоимости авиабилетов до Лондона, проживания в отеле Хилтон и регистрационного взноса, составляющего около 2 000 долларов. Налицо типичный при нынешнем состоянии дел пример отсутствия информации из-за отсутствия спроса, вызванного отсутствием денег.

Конференция проходила непосредственно в отеле Хилтон аэропорта Гэтвик и состояла в основном из семинаров, которые вели представители различных английских и американских компаний, занимающихся разработкой программных продуктов и обучением, в частности специалисты из AT&T Bell Laboratories, Hewlett-Packard и других фирм. Всего было прочитано 42 доклада, охватывающих большую часть проблем, касающихся практического применения языков C и C++ при разработке программных систем. Пожалуй, наибольший интерес среди участников вызвали лекция Бьярна Страуструпа, прилетевшего на пару дней из Нью-Йорка, и презентация фирмы Borland, на которой ее вице-президент Юджин Вэнг демонстрировал возможности нового пакета Borland C++ 3.1, поступившего в июне в продажу. Два дня работала небольшая выставка, где были представлены около двадцати различных фирм, в том числе Borland, Microsoft и Symantec, а также ряд других (в основном английских компаний). Поражает огромный объем литературы по программированию, представленной на выставке, причем не справочной, а методической, в которой, как правило, дается описание технологии разработки программного обеспечения для различных областей применения: баз данных, пользовательского интерфейса, сетей, численного анализа и т.д. Программирование постепенно становится индустрией, в которой появляются отдельные отрасли, идет создание новых технологий и, что не менее важно, растет информационная поддержка, обеспечивающая массовое распространение ноу-хау среди разработчиков и менеджеров. И если в ближайшие несколько лет российские специалисты по-прежнему будут лишены возможности знакомиться с новейшей литературой, мы рискуем безнадежно отстать в сфере разработки программного обеспечения. Реально может возникнуть ситуация, когда России придется импортировать не только компьютеры, но и специалистов, умеющих грамотно писать программы.

Цель подобных программистских форумов и чисто образовательная, и, в несколько меньшей степени, научная, а также рекламная. Под словом "рекламная" автор этих строк подразумевает не только рекламу фирм и их программных продуктов, но главным образом рекламу технологий программирования, новых идей и методов по созданию программного обеспечения. Дело в том, что сейчас на Западе быстрыми темпами идет переход этой отрасли на индустриальные рельсы, от кустарных методов к апробированным технологиям, облегчающим создание и сопровождение действительно сложных программных систем. Наиболее перспективным с этой точки зрения в настоящее время считается Объектно-Ориентированный Подход, предполагающий, что при написании программ в качестве основных абстракций используются не переменные, а объекты — структуры, содержащие как данные, так и функции, оперирующие этими данными. Каждый объект является относительно автономным образованием и описание его поведения можно рассматривать как отдельную работу. В результате появляется возможность разделять большую задачу на ряд менее сложных и решать их по отдельности. Но основная сила Объектно-Ориентированного Подхода заключается в возможности достижения гораздо большей степени повторного использования рабочих текстов программ, и, как одно из следствий этого, создания сложных программных систем, гораздо легче поддающихся модификациям. К сожалению, на русском языке выпущено пока очень мало соответствующей литературы, правда, скоро в издательстве "Мир" выйдет книга Грэйди Буча "Объектно-Ориентированное Проектирование с примерами применений", которая среди разработчиков, использующих ООП считается чуть ли не библией. Книга вышла в США в прошлом году и уже стала бестселлером.

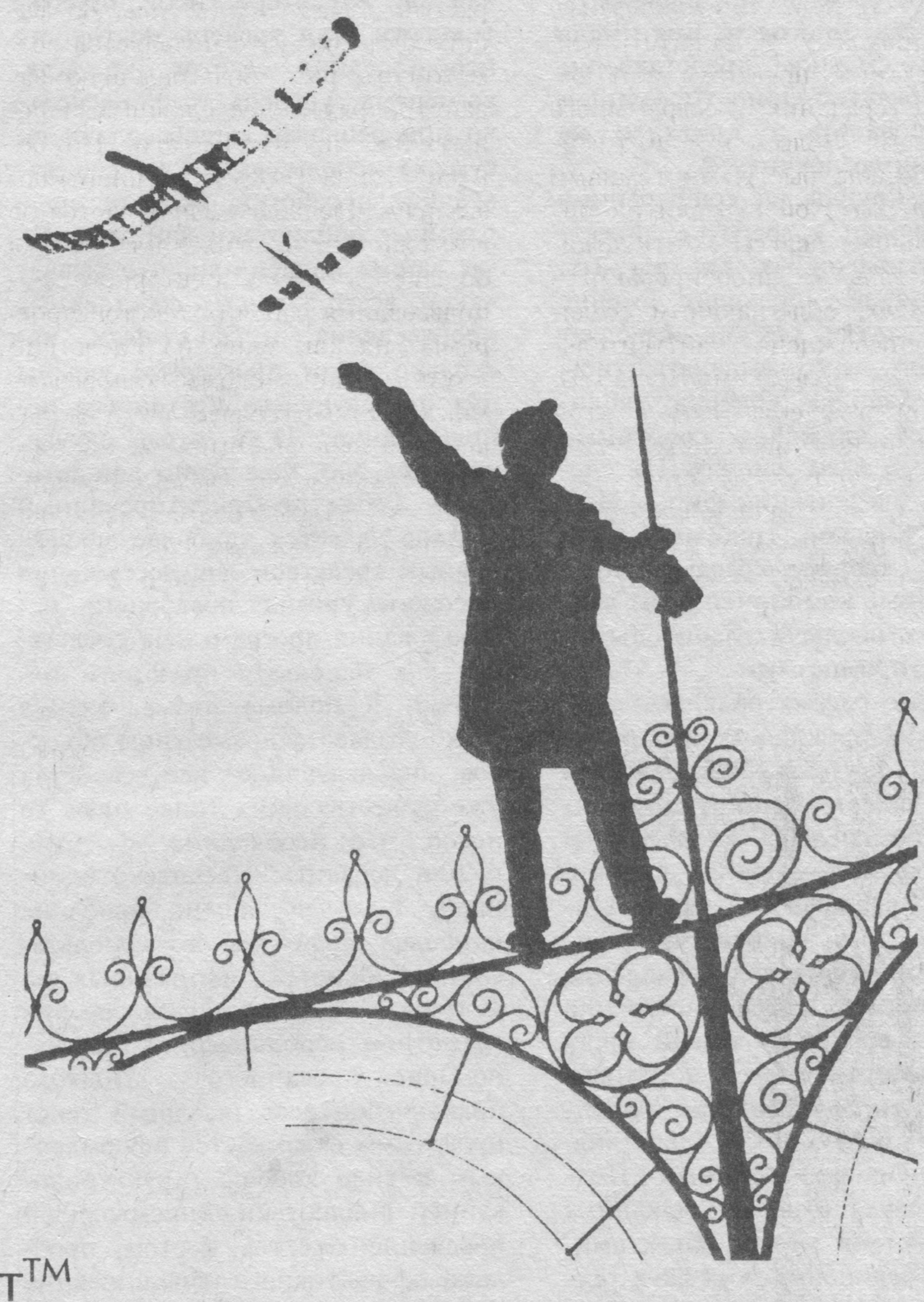


С++ является, наверно, самым распространенным объектно-ориентированным языком программирования, поэтому много внимания на конференции было уделено Объектно-Ориентированному Подходу. Сейчас одна из основных задач заключается в переводе огромной массы инженеров-профессионалов, использующих язык С в своих разработках, на использование С++ и в параллельном обучении их тонкостям ООП. Нет ничего удивительного в том, что значительная часть лекций и семинаров была посвящена именно этой проблеме. С++, вообще говоря, можно рассматривать просто как расширение языка С и писать на нем программы, не создавая объектов или используя их лишь в качестве вспомогательных инструментов. Но при этом основные преимущества ООП останутся нереализованными. Объектно-Ориентированное Проектирование требует от менеджеров и разработчиков выработки "нового мышления", основанного на способности оперировать абстракциями более высокого уровня. Для этого требуется определенная подготовка. Сейчас на Западе появилась масса фирм, специализирующихся на консалтинге и на обучении новым технологиям программирования. На конференции был представлен ряд таких фирм: Instrumatic, Semaphor Training, Object Designers, Saks & Associates, Insight Resource и другие. Выходит также ряд журналов, в той или иной мере посвященных Объектно-Ориентированному Подходу, например, С++ Journal, Journal of Object Oriented Programming, С++ Report, Object Magazine.

Надо сказать, что на семинарах по ООП излагались, как правило, основы метода, лекторы

# Элегантное решение для тех, у кого 386-е потребности и 286-ой бюджет

Компьютер
Саммит Элегант™ - это Ваш взлет на уровень современного программного обеспечения. Высокоинтегрированный дизайн фирмы CHIPS & Technologies обеспечивает высокую производительность и потрясающую надежность. В самом деле, кто бы иначе сопроводил его 2-летней гарантией!
Саммит Элегант™ не повредит Вашему зрению благодаря прекрасным характеристикам своей видеосистемы, скорость жесткого диска не заставит Вас скучать в ожидании загрузки программы или данных, а устройство защиты питания сделает вашу жизнь спокойной даже в условиях перепадов напряжения.



## Саммит Элегант™

- ❑ 386sx процессор 20 МГц
- ❑ Жесткий диск 52Мб (12 мс)
- ❑ Русифицированная клавиатура
- ❑ SuperVGA карта (1024x768)
- ❑ Лицензионная ДОС
- ❑ 2 года гарантии

SUMMIT SYSTEMS

Наши телефоны: Москва 299-1162 Минск 973-119

подробно объясняли, что такое объект, классификация и иерархия объектов, в чем заключаются свойства инкапсуляции, полиморфизма, как использовать виртуальные функции. Уровень подготовки слушателей оказался не столь высок, как можно было бы предположить. В частности, многие из них имели довольно смутное представление об Объектно-Ориентированном Проектировании и приехали на конференцию поучиться.

Ряд семинаров на конференции был посвящен разработке объектно-ориентированных баз данных. ООП наиболее подходит именно для этой задачи, так как позволяет достаточно легко выделять типы объектов (запись, колонка, таблица и т.д.), служащих основными элементами базы данных. Не случайно на презентации фирмы Borland с "высокой трибуны" было заявлено, что все создаваемые и планируемые компанией базы данных будут исключительно объектно-ориентированными.

Отдельно рассматривалась задача создания прикладных пакетов в средах Windows 3.X. Windows сама по себе является объектно-ориентированной средой, все элементы ее интерфейса — окна, кнопки, курсор — реализованы как объекты. Кроме того, архитектура Windows организована под управление прерываниями: среда постоянно посылает сообщения вашей программе, которая может в зависимости от типа сообщения выполнять те или иные действия. Объектно-Ориентированный Подход реализует сходный механизм взаимодействия между объектами, поэтому соединение этих двух технологий кажется достаточно перспективным. В настоящее время программные продукты под Windows постепенно вытесняют аналогичные системы, реализованные в DOS (достаточно сказать, что все, представленное на выставке, работало в среде Windows).

Как следствие применения более высоких технологий проектирования, по другим критериям стала оцениваться и квалификация программиста. Если раньше хорошим инженером мог считаться тот, кто умел за наиболее короткий срок написать наиболее сложную программу, то сейчас этого уже недостаточно. Конечно, умение быстро писать программы по-прежнему в цене, но не менее важной характеристикой продуктивности стал уровень повторного использования текстов, то есть, во-первых, уровень того, насколько при решении вашей задачи вы сумели использовать уже существующие наработки, будь то программные библиотеки или программы ваших коллег или, что бывает чаще всего, ваши собственные произведения, и, во-вторых, насколько ваши программы удобны для последующего повторного использования. И в этом случае, как, наверно, уже легко догадаться, Объектно-Ориентированный Подход является наиболее эффективным средством для достижения высокого уровня повторного использования программных средств, так как используя принципы иерархии и полиморфизма, можно легко создавать новые типы объектов, наследующие все свойства уже существующих типов плюс то новое, что необходимо изменить и/или добавить. Несколько семинаров было посвящено проблеме наиболее эффективного использования библиотек, написанных на С и на С++. Вообще говоря, при грамотном использовании Объектно-Ориентированного Подхода практически весь исходный текст программы оказывается оформленным в виде хорошо структурированной библиотеки, описывающей поведение объектов, поэтому проблема эффективного использования библиотек становится одной из наиболее важных.

На семинарах рассматривались также вопросы, касающиеся использования указателей, свойств типирования и иерархии, обсуждались проблемы, возникающие при создании больших программных комплексов, совместимых с несколькими операционными системами.

В заключение хотелось бы отметить прекрасный уровень организации конференции. Участникам еще до ее открытия были выданы конспекты почти всех лекций и семинаров, а также вся необходимая дополнительная информация о программе форума. Конференция проходила практически без сбоев, присутствовали все лекторы из тех, что были заранее объявлены, семинары и лекции начинались вовремя, причем параллельно велось по 4-5 занятий, и каждый слушатель имел возможность выбрать то, что ему наиболее полезно было бы узнать.

Следующая конференция под эгидой Бостонского университета, "Object Oriented Programming", состоится в Лондоне с 30 ноября по 4 декабря 1992 года. Заинтересованные лица и организации могут связаться с ее устроителями по телефону (044) 071-259-2032 или по факсу (044) 071-373-2067, а с автором этой статьи — по телефону (095) 461-61-28 или по электронной почте soft@stins.msk.su.

*М.Щелкин*



# К вершинам МАСТЕРства

Многим работающим с текстами на русском языке хорошо известен текстовый процессор ЛЕКСИКОН — первый и, как это ни невероятно, все еще единственный промышленный редактор текстов отечественной разработки. В своих постоянно обновляющихся версиях он легально и нелегально распространялся на российском рынке задолго до того, как сам этот рынок стал разрешенным в этой части света. Несколько менее известна (не более 100 тысяч легальных копий) интегрированная система МАСТЕР, сделанная на базе ЛЕКСИКОНа, с добавлением к нему электронных таблиц, деловой графики, системы ведения баз данных и встроенного языка программирования 4-го уровня.

Для людей, работающих на персональных компьютерах в области создания и использования систем автоматизации учреждений, и в особенности, для тех из них, кто уже ориентирован в своей технологии на инструментальную систему МАСТЕР, должно быть интересно взглянуть в ретроспективе и в перспективе на путь развития этой системы и поддерживаемой ею технологии.

## Вначале была наука

В то время, когда государство еще делало вид, что поддерживает науку, в одном из ее институтов — Вычислительном центре АН СССР, вопреки всему царила замечательная атмосфера настоящего творчества.

Школа глубоких ученых Н.Н.Моисеева, В.М.Курочкина, Ю.Г.Евтушенко, Ю.И.Журавлева, Г.С.Поспелова и многих других, хотя и не имела прямого отношения к таким пустякам, как редактирование текстовых файлов, формировала у своих учеников стиль и вкус, настраивала на наиболее фундаментальный уровень работы. В то же время более современные научные руководители В.П.Мазурик, В.М.Брябрин создали молодые (тогда) коллективы, которые первыми в стране начали осваивать персональные компьютеры (тогда еще всего лишь PC, а не PC AT и даже не PC XT).

Именно там были заложены основы оригинальных технологий, одна из которых к настоящему времени развилась в направление, представляемое промышленной системой МАСТЕР.

Первоначально разработки возникли без каких-либо серьезных планов. Просто мы редактировали тексты с помощью текстового редактора АБВ (А.Борковского), который был очень неплох, но не был лишен недостатков. И потому захотелось написать что-нибудь получше только для того, чтобы легче было вводить и редактировать собственные программы (вычислительные) и статьи (научные). Получился редактор "Е9", названный так потому, что в нем было 9 окон, и превратившийся потом в ЛЕКСИКОН путем добавления десятого окна и меню в стиле Lotus 1-2-3. И почти одновременно с рождением текстового редактора стала появляться вторая программная система — МАСТЕР, которая и должна была служить технологической основой для интеграции разных видов информации: начиная с текстов, таблиц и графиков и кончая информационными, математическими и конструкторскими моделями.

## Что получилось: средо-ориентированная технология

Если ЛЕКСИКОН — это всего лишь простой, удобный и надежный текстовый процессор, полезный именно тем, что не требует от пользователя усвоения каких бы то ни было моделей и технологий, то МАСТЕР обладает определенными уникальными характеристиками. Он индуцирует и поддерживает особого рода технологию разработки прикладных систем, названную "средо-ориентированной".

Главная идея проста (стремление к простоте и фундаментальности понятий унаследовано в МАСТЕРе от ЛЕКСИКОНа): разработку прикладной системы следует проводить не каким-то сторонним инструментальным средством, а непосредственно изнутри разрабатываемой системы, находясь в ней как в среде, снабженной средствами саморазвития. При этом не просто существенно укорачивается рабочий цикл "идея-программа-проба", но, что более важно, у разработчика исчезает ощущение отчужденности инструмента от объекта разработки. Разработчиком может оказаться конечный пользователь, который, даже и не подозревая о том, что занимается программированием, просто постепенно обустраивает свой электронный рабочий стол. Опыт использования МАСТЕРа показывает, что именно это качество наиболее важно для конечного пользователя, который не хочет отвлекаться от своих проблем на освоение каких бы то ни было новых инструментальных средств.

## Как это делается

Способ реализации этой идеи таков. Система МАСТЕР в своей основе — универсальная информационная среда, ни на что заранее не ориентированная, но обладающая многочисленными возможностями структурного представления информации, ее диалоговой презентации и редактирования, алгоритмической обработки и долговременного хранения в больших объемах.

В этой среде, изначально пустой, как поверхность нового письменного стола, вы можете создать и нужным образом разместить те или иные информационные объекты — тексты, таблицы, графики. Их можно комбинировать друг с другом, образуя сложные документы, с помощью составных рамок. Кроме того, между этими объектами можно устанавливать связи алгоритмического характера с помощью встроенного языка программирования.

Как выяснилось из практики, существенно то, что язык имеет русские мнемоники для всех функций и ключевых слов. Будучи принципиальным противником попыток "русификации" стандартных чужеродных языков программирования (такие попытки в свое время делались для языка АЛГОЛ), я, к своему удивлению, обнаружил, что для системы программирования 4-го поколения, пользователями которой являются непрограммисты, и в которой язык программирования играет не первостепенную роль, а всего лишь вкладывается внутрь информационной системы, программы на котором совершенно перемешаны с собственно обрабатываемыми данными, так вот, для такой системы встроенный язык программирования воспринимается намного естественнее, если по лексике он совпадает с той информацией, внутри которой "живут" его программы.



Итак, МАСТЕР — это универсальная среда с информационными объектами разных типов и с объединяющим их встроенным языком программирования. В таком виде МАСТЕР использовался многочисленными разработчиками и конечными пользователями для решения самых разных задач — от автоматизации бухучета до статистического моделирования.

## Что доступно пользователю сейчас

К настоящему времени полностью сформировалась система технологического развития и поддержки обеих программных систем — ЛЕКСИКОНа и МАСТЕРа. Обе программные системы доступны в коммерческом виде как для поставок, так и для получения лицензий на распространение.

В то же время коллектив разработчиков занимается созданием нового поколения интегрированной средо-ориентированной технологии, в которой и ЛЕКСИКОН, и МАСТЕР получают второе дыхание, развиваясь в самых различных современных направлениях.

## Что же будет?

Очень непросто в мире современного программного обеспечения, переполненном всевозможными яркими изделиями, отыскать свое лицо. Непросто, но и совсем не сложно. Особенно на нашем рынке со всеми его контрастами, где новейшие компьютеры соседствуют не только с "Искрами" и ЕС, но с обычными счетами, где все знают про локальные и про глобальные спутниковые сети, но не могут дозвониться на соседнюю АТС. Эти контрасты упомянуты вовсе не для красного словца — они порождают действительную проблему в отношении использования здесь, на нашей почве, любой из замечательных западных технологий даже после успешной вставки в них русских букв. Эти технологии создавались для совсем иной инфраструктуры.

В этой ситуации представляется очень желательным дать пользователю инструмент, с помощью которого он мог бы разрабатывать системы (с одной стороны, способные работать на маломощных компьютерах класса ЕС1841, но с другой стороны, имеющие внутреннюю архитектуру современного типа, готовую для безболезненного переноса в сложные компьютерные системы графического и сетевого характера).

Технология МАСТЕРа-2 разрабатывается сейчас именно таким образом. Итак, основные новые черты этой технологии:

- модульное программирование и открытость системы на уровне языков Си и Си++ на основе техники DLL (динамически компонуемых библиотек);
- объектно-ориентированное представление всех свойств информационной среды, включающее иерархию классов, замыкание свойств, виртуальные функции;
- событийно-управляемая организация интерфейса с поддержкой мыши, таймера и любых других устройств;
- расширенная модель текстов, содержащая различные языки, различные шрифты, графические вставки, сложную структуру текстового поля;
- виртуальная память как для программ, так и для данных, работающая прозрачно на расширениях EMS, XMS и на дисковых файлах;
- поддержка всех элементов современного пользовательского интерфейса — свисающие меню, всплывающие диалоговые панели, кнопки, рамки;
- независимость технологии от используемой оконной платформы (DOS, WINDOWS, PM OS/2, X Windows) за счет использования абстрактных классов диалоговых объектов, а между MS-DOS и WINDOWS и более того — за счет совместимости по формату DLL-модулей.

Последнее свойство особенно важно для текущего состояния российского рынка программ. Оно позволяет разрабатывать версии прикладных систем для маломощных конфигураций и использовать их в обычном MS-DOS, а затем без каких-либо изменений (разве что с добавками, использующими новые мощности) переносить под MS WINDOWS.

Помимо всех этих новых черт основой технологии МАСТЕРа-2 по-прежнему остается ее средо-ориентированный характер. Впрочем, добавление явной объектной ориентированности (неявная имеется и в МАСТЕРЕ-1) придает этой технологии новое качество. Теперь можно не просто добавлять к среде какие-то новые элементы и связи, но и определять в виде прототипов новые классы этих элементов (т.е. классы типовых диалоговых сред). Это позволяет говорить о технологии как о визуальной раскрутке, подобной раскрутке компиляторов, когда элементы языка, только что реализованные на компиляторе без этих элементов, немедленно вставляются в раскручиваемый компилятор для порождения очередного слоя концептуальных элементов.

И самым первым шагом всего дерева будущих виртуальных раскруток является создание в новой инструментальной среде текстового процессора ЛЕКСИКОНа-2.

## Сейчас

В мае 1992 года подготовлен к использованию первый инструментальный слой системы, на котором средо-ориентированная технология визуальной раскрутки может оттачиваться в Мастер-центре фирмы "Микроинформ" и изучаться всеми разработчиками, которые желают ознакомиться с ней, не дожидаясь официального промышленного выпуска. С этой целью в "Микроинформе" периодически собираются учебные группы, участники которых в конце обучения получают бета-версию инструментария МАСТЕРа-2 (уровня языка Си).

*Е.Веселов*

# Компьютеры LAND

Множество деловых и псевдоделовых людей сегодня все еще занимаются или пытаются заниматься компьютерным бизнесом. Информация о "живых" или "контрактных" машинах передается "с глазу на глаз" под честное слово, между реальным продавцом и потенциальным покупателем выстраивается целая вереница посредников, и хвост ее теряется в закоулках секретности. Результаты подобной коммерческой активности, как правило, равны нулю. Даже если случится невозможное, и к покупателю приходит, наконец, долгожданная драгоценная техника, вопрос о ее гарантийном или послегарантийном обслуживании так и остается нерешенным.

Качество приобретаемых таким образом "персоналок"..., а вообще, стоит ли здесь говорить о качестве? Несмотря на обилие разнообразных сертификатов, фирменных наклеек и письменных гарантий чистоты кровей, компьютеры отказываются выполнять даже стандартные тесты. Происходит это, очевидно, потому, что разнородные комплектующие этих неизвестно где, кем и в каких условиях сработанных машин постоянно конфликтуют друг с другом.

Подобные неприятности никогда не происходят с покупателем, который имеет дело с солидной фирмой. Но компьютеры с маркой Hewlett-Packard, Zenith или Compaq дешево не купишь, а приобретать традиционно высококачественное и, соответственно, дорогое оборудование наши бизнесмены пока еще не научились. Получается порочный круг: или отлично работает, но "кусается", или и не кусается и не работает, зато по-дешевке.

Есть, правда, один выход — довериться какой-нибудь молодой российской или совместной фирме, специализирующейся на сборке и обслуживании персональных ЭВМ. Поговорим об одной из таких фирм.

Знакомьтесь — московская компания LAND, а ей не так давно исполнилось 3 года, — акционерное общество закрытого типа с уставным капиталом 50 млн. рублей. Фирма выгодно отличается от сверстников прежде всего тем, что практически не занимается "чистой коммерцией", а специализируется на сборке и комплектации вычислительной техники.

Приятно поражает западный стиль работы компании. Здесь стоит упомянуть и доставку аппаратуры непосредственно покупателю в любую точку России, и систему поощрений, принятую для оптовых и постоянных клиентов, и 45 часов, за которые фирма производит ремонт вышедшего из строя оборудования. Даже наличие платной автостоянки рядом с офисом выгодно отличает LAND от конкурентов.

Но все-таки главное преимущество компании LAND — это высокое качество и богатство ассортимента производимой продукции. За все время существования фирмы она не получила ни одной рекламации. Что же касается ассортимента, то здесь мы видим практически всю гамму IBM-совместимых персональных компьютеров, оснащенных надежными "винчестерами" Seagate, быстрым ОЗУ (70 нс) и кэш-памятью объемом до 256 Кбайт. В комплекте с цветными SVGA-мониторами и матричными принтерами ведущих фирм, таких как Star, Epson и Okidata, компьютеры с маркой LAND сегодня оказались практически самыми доступными на внутреннем рынке России.

А всего четыре года назад весь штат фирмы состоял из учредителей — четырех молодых инженеров, начавших свою коммерческую деятельность с разовых поставок в Союз буквально единичных экземпляров бытовых электроприборов и персональных компьютеров.

Несмотря на столь скромный старт, нынешний оборот компании составляет около одного миллиарда рублей в квартал. Сотни квалифицированных специалистов в цехах фирмы производят сборку самых современных настольных персональных компьютеров — от недорогой "рабочей лошадки" AT-286, до высокоскоростных графических станций на базе микропроцессора Intel 486/50. Стоит упомянуть, что производственные площади компании LAND находятся не только в России, точнее — в Москве, но и в Сингапуре и в Австрии.

Комплектующие, необходимые для сборки 200-300 вычислительных машин в сутки, поступают на конвейеры LAND непосредственно от фирм-изготовителей, имеющих достаточно громкие имена и прекрасную репутацию на мировом рынке.

Но простейшая "отверточная" сборка компьютеров с некоторых пор перестала удовлетворять руководство LAND, так что в скором времени ее начнут вытеснять современные высокотехнологичные производства. Уже сейчас компания приступила к изготовлению портов ввода-вывода, а не сегодня-завтра вступит в строй завод по производству материнских плат. Кроме того, в ближайшие тактические планы фирмы входит строительство завода по производству матричных принтеров, совместимых с популярными печатающими устройствами Microline. Предполагается, что механические узлы должны поставляться из-за рубежа, а вся электроника и окончательная сборка устройств будут производиться на месте.

Разумеется, сфера деятельности компании не ограничивается только изготовлением компьютеров. Реализация комплектующих, продажа лицензионно-чистого программного обеспечения, а также гарантийное и послегарантийное обслуживание вычислительной техники — во всех этих смежных областях LAND пока не имеет конкурентов в России.

Стратегия руководства фирмы строится на том, что наиболее ощутимые результаты капиталовложений в нашей стране можно получить только от инвестиций в самые неблагополучные отрасли. В нашем конкретном случае на выбор сферы деятельности, очевидно, повлияло болезненное состояние российской информатики. Современные персональные компьютеры и высококачественное программное обеспечение, постоянная модернизация производства, замена устаревших моделей более производительными и переход на новейшие платформы — все это в конечном счете и составляет "линию жизни" компании LAND.

*Апо Лайм*

# Новости

Фирма Microsoft заявила о задержке с выпуском Windows NT (New Technology). Согласно более ранним заявлениям, система должна была появиться на рынке в конце 1992 года.

Новая дата — "начало 1993 года". Представители Microsoft заявляют, что "им нужно больше времени дабы правильно отреагировать на предложения пользователей об улучшении системы". В начале июля более 20 тысяч фирм-разработчиков программ получили для оценки software developers kit (SDK) нового продукта. Бета-версия Windows NT в настоящее время производится и будет разослана тем, кто удостоился чести ее тестировать, в середине октября.

Один из директоров Microsoft сказал, что точная дата начала поставок будет определена в ноябре-декабре. На выставке Comdex будет более 300 прикладных программ, рассчитанных на работу в среде 32-разрядных Окон.

*Newsbytes News Network, September 28, 1992*

Okidata выпустила настольный аппарат размером с лазерный принтер, совмещающий в себе факс, сканер, лазерный принтер и копировальную машину.

Устройство Doc-It выпущено в двух модификациях, которые работают вместе с любым компьютером под DOS или Windows. Doc-It 3000 имеет разрешающую способность в 300 точек на дюйм и использует систему команд с Hewlett-Packard HP-PCL4+. Doc-It 4000 работает с разрешением 400 точек на дюйм и поддерживает PostScript.

Устройство имеет размер 42.5x42.5x20 см. К персоналкам оно подключается посредством специальной сопрягающей карты, работающей как факс-модем и соединяющей все остальное с компьютером. Приходящие факсы могут сохраняться на компьютере или сразу распечатываться.

Рекомендованная цена на устройство с меньшей разрешающей способностью 3999 долларов, а на Doc-It 4000 — 4999 долларов.

*Newsbytes News Network, September 28, 1992*

Microsoft добавляет **звуковые возможности к Windows 3.1.** Они будут предназначены в основном для деловых людей, позволяя записывать голосовые сообщения и воспроизводить их. Говорят, что этот режим будет очень полезен при проверке (методом вычитывания) многочисленных цифровых данных, введенных в электронную таблицу.

Пользователь также сможет добавить небольшое голосовое сообщение к исполняемому файлу, типа строгого указания начальнику исправить отмеченные в документе недостатки.

Windows Sound System будет дополнением к основному комплекту Windows и начнет продаваться со следующего месяца. Устройство, состоящее из платы расширения, устанавливаемой в компьютер, микрофона и наушника будет стоить 289 долларов.

Apple, Compaq и Next уже снабжают свои машины оборудованием для записи и воспроизведения речи. Приобретение машин этих фирм избавит вас от необходимости добавления специальной платы. Правда придется купить сами машины — весьма недешевые.

Представитель Microsoft заявил, что использующийся для воспроизведения звука режим OLE (object linking and embedding) поддерживается сейчас более чем 150 прикладными программами, в числе которых PowerPoint, Micrografx Designer, Norton Desktop и Wordperfect for Windows.

*Newsbytes News Network, September 21, 1992*

В шуме, сопровождавшем объявления о скором появлении в продаже операционной среды Windows for Workgroup фирмы Microsoft, остались незамеченными несколько интересных заявлений руководителя фирмы.

Наши читатели вероятно уже слышали, что компания продемонстрировала рабочую версию **базы данных Access** (ранее называвшуюся Cirrus и имевшую секретных названий больше, чем Италия правительств), активно говорит о версии FoxPro для Windows и показала несколько продуктов для работы в Windows for Workgroup.

Но те, кто долгое время наблюдают за фирмой, не пропустили комментарий Билла Гейтса, председателя и сооснователя фирмы, высказанный им в ответ на недавние сообщения о том, что Федеральная Торговая Комиссия собирается предать гласности некоторые данные ее 18-месячных изысканий торговой практики крупнейшей программистской компании.

Статья в BusinessWeek породила слухи о том, что **Microsoft может быть**, как и десятилетие назад AT&T, **расформирована** по решению суда на несколько меньших, независимых компаний.

Гейтс сказал, что все эти слухи неверны, и что члены Федеральной комиссии все еще только готовят свое подробное сообщение, и не в их традициях сообщать прессе что-то до окончательного утверждения доклада.

Гейтса спросили также о его знаменитой памятной записке персоналу фирмы, в которой он высказывал боязнь того, что может случиться "самый кошмарный сценарий" развития событий — все конкуренты фирмы, включая IBM и Apple, объединяются против Microsoft. По заявлению председателя, он и в этом году в послании к своим сотрудникам напишет что-то похожее.

"Всегда есть риск, что люди научатся писать программы быстрее, чем мы", — сказал Гейтс. — "Я буду продолжать указывать на то, в какой ситуации мы оказываемся, но без излишних пугающих указаний. Я также буду говорить о том, что мы должны двигаться вперед еще более быстрыми темпами, дабы успешно конкурировать с многочисленными маленькими компаниями, работающими на данном рынке".

*Newsbytes News Network, October 6, 1992*

**Everex** заявила, что старается быть конкурентоспособной, снижая цены на компьютеры и снижая себестоимость. Сообщено, что цены на некоторые модели уменьшены на 59 процентов, и что примерно 250 человек уволены. Между объявлением о планируемых увольнениях и собственно сокращением штата до 2200 сотрудников прошло всего три дня.

Снижение цен достигает 1000 долларов для некоторых моделей. Например, компьютер-блокнот 386SX Tempo Carrier с 8 Мбайтами памяти и 80-Мбайтным винчестером и факс-модемом теперь стоит 2199 долларов. Подешевели и другие модели компьютеров и периферии типа модемов, сетевых карт и стримеров.

В начале сентября фирма объявила о начале выпуска компьютеров с процессором Intel 486DX2-66. Это сетевой файл-сервер, пара настольных моделей и Tower.

А еще весной Everex собирался купить Northgate, крупного производителя компьютеров, но не сделал этого, "...ощущая, что Northgate имеет слишком много долгов," — сказал представитель фирмы.

*Newsbytes News Network, September 28, 1992*

В сентябре 1992 года фирма **Intel** начала проводить серию семинаров, первый из которых состоялся в одном из уютных подмосковных домов отдыха. На семинар были приглашены представители крупнейших московских предприятий и организаций. Проведение серии семинаров преследует несколько целей, одной из которых является привлечение широкого круга разработчиков и потребителей к изделиям, выпускаемым Intel.

На семинаре сотрудники фирмы подробно рассказали о том, что же представляет собой корпорация Intel осенью этого года. Большое внимание было уделено микропроцессорным комплектам и микропроцессорам "второй волны", а также средствам мультимедиа. В фойе конференц-зала была организована небольшая выставка компьютеров, компонентов для коммуникаций и других технических средств, которые фирма Intel предлагает на отечественный рынок. К слову сказать, устроители семинара побеспокоились не только о пище духовной: в перерывах можно было не только перекурить, но и закусить, ну а обед вообще был достоин высших похвал. Подобные семинары (за качество сервировки стола трудно ручаться) пройдут также в городах Харьков, Минск, Киев, Рига и Санкт-Петербург. Организатором семинара в Москве была фирма А/О Стинс Коман, являющаяся официальным дистрибьютором Intel.

*КомпьютерПресс*

Компания **Ncube** заявляет, что массивный параллельный компьютер еще не столь мал, чтобы ставить его на стол, но стоять около стола он уже вполне может. Фирма предлагает желающим начать работать с суперкомпьютерами серии мини-суперов 2E.

Ncube заявляет, что используя принцип параллельных вычислений, фирма сделала машину, которая не дороже хороших рабочих станций или мини-машин (цена от 29950 долларов).

В полной конфигурации машина имеет производительность до 1280 MIPS (миллионов инструкций в секунду) и 422 MFLOPS (миллионов операций с плавающей точкой в секунду). Система может иметь от 8 до 128 процессоров, от 32 Мбайт до 4 Гбайт ОЗУ и до 24 Гбайт дисковой памяти, и все это стоит около вашего стола...

Фирма заявляет, что машина очень хорошо ведет себя как сервер баз данных. Oracle поддерживает примененный в машине параллельный подход.

Машины полностью совместимы с более мощными компьютерами этой же фирмы. Программы, написанные для 8-

процессорного Ncube 2E с тем же успехом выполняются и на Ncube 2S с 8192 процессорами, 512 Гбайтами ОЗУ, производительностью 34 GFLOPS и ценой примерно 1000 долларов за 1 MFLOPS производительности.

*Newsbytes News Network, July 14, 1992*

### WAVEGUIDE — существенное нововведение в области беспроводной связи

Компания Motorola выпустила устройство WaveGuide, которое может соединить любой компьютер с любой беспроводной системой передачи данных. WaveGuide принят на вооружение службой радиопередачи данных Ardis, являющейся совместным предприятием Motorola и IBM, и, вероятно, будет использоваться и ее конкурентом — службой RAM Mobile Data.

Развитие служб сотовой и радиопередачи данных сдерживается отсутствием стандартов. WaveGuide будет предложен всем, кто захочет им воспользоваться. Если использовать WaveGuide повсеместно, то разработчикам прикладных программ не придется выяснять особенности сети перед тем, как начать писать программу.

Вероятно, в будущем компьютеры будут постоянно включены во многие сети, обмениваясь данными с ними.

Сюзан Мейджор (Susan Major), управляющая разработкой программного обеспечения для WaveGuide, сказала, что работает над драйверами WaveGuide как для пакетов программ, так и для сетей. К июлю появится версия для Macintosh, за ней последуют версии для Unix и MS Windows. Работа над созданием "библии" для WaveGuide ведется вместе с компаниями, занимающимися разработкой программного обеспечения.

*The Teleputing Hotline, February 10, 1992*

### America Online предлагает выставку советских секретов

Сеть America Online совместно с Библиотекой Конгресса США открыла он-лайновую выставку. Библиотека по соглашению с Комитетом по делам архивов Российской Федерации предоставит наиболее сенсационные материалы ранее засекреченных советских архивов под названием "Откровения из русских архивов". Выставка, содержащая отрывки из более чем 300 документов, будет работать одновременно с более обширной экспозицией документов, которая откроется 17 июля в здании Библиотеки Конгресса (Мэдисон).

Выставка довольно интересная — знаете, этакое достаточно взвешенное описание нашей истории в файле размером 200 Кбайт. Единственным минусом является буквальный — слово в слово — перевод всего на английский язык. Это мы с вами знаем, что означает фраза "Копия верна" снизу на ма-

шинописной странице. Переведенная на английский язык, она выглядит както очень загадочно...

И еще. Нигде никто не говорил, сколько и кто на этом деле заработал.

*The Teleputing Hotline, June 22, 1992*

Фирма **HYUNDAI** будет перепродавать технологию Personal Xchange, которая позволяет одновременно передавать по одной линии голос, данные и изображение, компании First Pacific Networks.

Компании **MATSUSHITA** и **SOUTHWESTERN BELL** совместно разработают карманный телефон, который можно будет использовать и в автомобиле, и как обычный телефон в офисе.

Фирмы **TRAVELING SOFTWARE** и **SLATE** представили 3 пакета программ связи для компьютеров с рукописным вводом информации, которые позволяют осуществлять передачу и совместное использование данных вне зависимости от базовой платформы.

*The Teleputing Hotline, February 17, 1992*

**Два** молодых компьютерных **"хакера"** (взломщика) **арестованы** в Нидерландах за несанкционированный доступ к сетям по всему миру. В прошлом эту страну критиковали за несоблюдение законодательства о компьютерной безопасности. Похоже, что этим двоим будет предъявлено обвинение в подделке, компьютерном мошенничестве и нанесении убытков криминальными действиями.

*The Teleputing Hotline, February 10, 1992*

Более 6000 домовладельцев Австралии получили официальные письма, содержащие личные и финансовые сведения о других людях. То, что должно было быть обычным сообщением Министерства Социального Обеспечения о пособии на детей, было разослано по неправильным адресам, в результате некоторые люди получили подробную информацию о банковских счетах соседей. Это не первый случай такого рода. **Назревает большой общественный скандал** по поводу быстрого увеличения объема информации, хранимой в центральном компьютере в столице Австралии, Канберре.

*The Teleputing Hotline, February 10, 1992*

Полиция штата Нью-Йорк образовала **отдел по борьбе с компьютерной преступностью** под руководством Рональда Стивенса (Ronald R. Stevens). Начальные расходы частично возместило федеральное правительство.

ZIFF будет обеспечивать соединения между службами Dow Jones News Retrieval и Market Identifiers компании Dun & Bradstreet, а также со службой SEC Online. Все это работает на оборудовании DEC VAX.

*The Teleputing Hotline, February 10, 1992*

## КАНАДА: Биржа Торонто закрывает торговый зал

Грант Баклер сообщает Newsbytes, что члены Фондовой Биржи Торонто (TSE), которая является крупнейшей в Канаде, проголосовали за полный переход к автоматизированной системе торгов. Торговля акциями в торговом зале будет прекращена к концу марта 1993 года после 114 лет торгов в зале. Сделки с опционами и фьючерсами попрежнему будут проводиться в открытом торговом зале.

Директора TSE считают, что это лучший способ конкурировать с более крупными биржами США. Сейчас в торговом зале TSE работает около 350 человек, некоторые останутся у терминалов, но примерно сотне клерков, видимо, придется уйти. Система Автоматизированной Биржевой Торговли (Computer-Assisted Trading System (CATS)) торонтской биржи была одной из первых таких систем в мире. Аналогичные системы были проданы некоторым другим биржам.

*The Teleputing Hotline, February 17, 1992*

## Программируйте по стандарту

Ура! Наконец-то кончилась для наших программистов пора блужданий в потемках и изобретаний американского велосипеда на российский лад. Речь идет о доморощенном выдумывании и изготовлении многочисленных вариантов пользовательского интерфейса прикладных программ.

Теперь каждый отечественный умелец может проектировать интерфейс своих программ на уровне мировых стандартов, а точнее, стандарта Common User Access фирмы IBM, который разработан в рамках известного проекта SAA (Systems Application Architecture) и является сегодня общепринятым у разработчиков программного обеспечения.

Прочитать об этом стандарте вы можете в новой книге с длинным названием "Проектирование пользовательского интерфейса на персональных компьютерах. Стандарт фирмы IBM", под редакцией М.Дадашова, выпущенной литовской фирмой DBS Ltd.

В книге описаны составные части интерфейса и варианты их реализации, способы организации диалога с пользователем. Книга снабжена множеством рисунков и справочных таблиц, а также глоссарием.

*К.Чащин, А.Синев, А.Борзенко*

# ВСЕ ДЛЯ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ IBM PC/XT/AT !

## ПОСОБИЯ ДЛЯ РУКОВОДСТВА В ЕЖЕДНЕВНОЙ РАБОТЕ И ОБУЧЕНИЯ ПЕРСОНАЛА

*МОСКВА, 129010, а/я 837, МП "РОСК". Телефоны: (095) 168-53-06, 475-89-84 с 14 до 17 часов.*

**КНИГА "РЕМОНТ И МОДЕРНИЗАЦИЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ"**
В книге объемом 690 стр. подробно описана установка, конфигурирование, эксплуатация и ремонт компьютеров семейства IBM PC/XT/AT/PS/2. Книга полезна в качестве руководства при ремонте и обслуживании системы, а также в качестве справочника для понимания взаимодействия и работы компонентов системы.
Стоимость книги — 2300 руб. + 28%.

**ПРАКТИЧЕСКОЕ ПОСОБИЕ-СПРАВОЧНИК "РЕМОНТ БЛОКОВ ПИТАНИЯ КОМПЬЮТЕРОВ, МОНИТОРОВ И ПРИНТЕРОВ"**
В книге подробно рассмотрена схемотехника блоков питания импортных IBM-совместимых компьютеров типа XT,AT,AT-386, видеомониторов CGA, EGA и VGA, а также принтеров. Приведены схемы более десяти блоков питания различных фирм-изготовителей. Изложены методики поиска и устранения типичных неисправностей блоков питания, а также рекомендации по замене импортных элементов отечественными. Книга окупается после первого ремонта блока питания.
Стоимость книги — 4000 руб. + 28%.

**ПРАКТИЧЕСКОЕ ПОСОБИЕ-СПРАВОЧНИК "ВИДЕОМОНИТОРЫ И ВИДЕОАДАПТЕРЫ EGA, VGA и SuperVGA"**
В книге подробно рассматриваются схемотехника видеомониторов и видеоадаптеров различных фирм-изготовителей — описание функционирования, сигналов, цоколевки микросхем, внутренняя структура, входные и выходные сигналы.
Стоимость книги — 3000 + 28%.

**Справочник STANDARD IBM PC**
Устройство, установка, техническое обслуживание и ремонт персональных компьютеров. Описания комплектующих компьютера IBM. Справочные сведения по винчестерам.
Стоимость справочника — 1800 руб. + 28 %.

**Справочник по стандарту шины ISA**
Сигналы, осциллограммы, назначение выводов и ПР. Незаменимое пособие для разработчиков дополнительных устройств к IBM PC.
Стоимость — 1000 руб. + 28 %.

**Справочник ROM BIOS**
Справочник по прерываниям стандартного BIOS IBM-совместимого компьютера.
Стоимость — 1000 руб. + 28%.

**ПРАКТИЧЕСКОЕ ПОСОБИЕ-СПРАВОЧНИК "Накопители на дисках в IBM PC XT/AT и их контроллеры"**
В книге описаны методы кодирования, записи и воспроизведения информации, предназначенной для хранения на НМД. Рассмотрены структурные и функциональные схемы контроллеров и накопителей на гибких и жестких дисках. Описаны ряд однокристальных контроллеров НГМД и НЖМД, другие СБИС, используемые для построения контроллеров. Рассматриваются основы организации размещения файлов на НМД, а также вопросы подготовки к работе и обслуживание НМД. Заключительная глава посвящена перспективам развития накопителей на дисках. В приложении приведены принципиальные схемы некоторых накопителей и контроллеров. Материал представляет интерес для широкого круга инженеров, занимающихся ремонтом и обслуживанием компьютеров IBM PC/AT. Стоимость — 3000руб + 28%.

**АЛЬБОМ СХЕМ КОМПЬЮТЕРОВ IBM PC/XT/AT**
Альбом содержит 30 схем компонентов компьютеров IBM :
- шесть схем материнских плат (2 XT и 4 AT);
- три схемы блоков питания (150, 180 и 200 Вт);
- четыре схемы мониторов (MONO, CGA, EGA, VGA);
- семь схем клавиатур;
- схемы адаптеров и контроллеров мониторов, дисководов, портов.

Схемы выполнены на 120 листах формата А4.
Стоимость альбома — 4500 руб.+28%.

**АЛЬБОМ ПРИНЦИПИАЛЬНЫХ СХЕМ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ ЛИНИИ IBM PC XT/AT/SuperAT-386**, включающий в себя: две различные схемы материнской платы IBM PC XT; восемь схем материнской платы IBM PC AT; три SuperAT-386; в том числе и собранных на сверхбольших чипах; различные схемы следующих устройств: адаптеров мониторов, контроллеров коммуникационных портов ввода-вывода, контроллеров HARD и FLOPPY дисков, аналоговые платы HARD и FLOPPY дисков, видеомониторов, клавиатур, манипуляторов "мышь", принтеров и блоков питания (всего 218 страниц формата А3).
Стоимость 7000 руб. + 28 %.

**СХЕМОТЕХНИКА IBM PC AT**
На примере классической схемы компьютера IBM PC AT подробно описано устройство систем этого класса. Рассмотрено подключение основных микросхем и функционирование шины стандарта IBM PC AT. Книга полезна при ремонте компьютеров AT и для разработки собственных плат расширения. Всего 96 листов формата А4.
Стоимость — 1000 руб.+ 28%.

**СПРАВОЧНИК ПО СВЕРХБОЛЬШИМ МИКРОСХЕМАМ ТИПА VLSI**
В справочнике приведены цоколевка, блок-схемы и внутренняя организация сверхбольших микросхем VLSI серии 100, 200 и 300.
Стоимость справочника — 2000 руб. + 28%.

**Справочно-учебное пособие по микросхемам комплекта INTEL**
В книге систематизирована информация по микросхемам применяемым в компьютерах совместимых с IBM PC/AT. Приведены цоколевки микросхем с описанием сигналов и комментариями. Рассмотрены внутренние блок-схемы, принципы функционирования, типовые включения. Описания работы иллюстрируются временными диаграммами.
Стоимость — 3000 руб. + 28%.

**Справочник по кодам ошибок самотестирования компьютеров ИБМ PC/XT/AT**
Справочник содержит расшифровку более 500 кодов и сообщений об ошибках выводимых на экран при самотестировании ПЭВМ-аналогов ИБМ PC/XT/AT. Поставляется на дискете.
Стоимость справочника — 1000 руб.+28%.

**Карты поиска и устранения неисправностей компьютеров ИБМ PC/XT/AT**
В пособии описаны пошаговые процедуры поиска и устранения неисправностей в компьютерах ИБМ PC/XT/AT. Поставляется на дискете.
Стоимость — 1000 руб.+ 28%.

**КОМПЛЕКТ ДИАГНОСТИЧЕСКИХ ПРОГРАММ ДЛЯ ИБМ-СОВМЕСТИМЫХ ПЭВМ**
Ремонтно-диагностические программы позволяют проверить исправность монитора, дисководов, винчестера, памяти, клавиатуры. Незаменимы при покупке и ремонте дорогостоящей техники.
Стоимость — 1000 руб.+28%.

**Техническое описание процессора Intel 80386** — 628 Кбайт. Стоимость — 1500 руб.

### Диагностическая плата "СЛОТ-ТЕСТЕР" для ПЭВМ типа IBM PC/AT-286

Совершенно новый, уникальный диагностический прибор. Диагностическая плата вставляется в свободный разъем расширения PC/AT и менее чем за одну минуту позволяет оценить исправность компьютера. Плата способна обнаруживать более 47 видов неисправностей:
- неисправности BIOS и CPU,
- ошибки памяти,
- ошибки видеопамяти,
- ошибки при обработке прерываний,
- неисправности канала прямого доступа,
- неисправности контроллера клавиатуры и многое другое.

Наличие на плате двух семисегментных индикаторов позволяет производить тестирование без дисплея. Диагностическая плата подробно описана в статье нашего ведущего специалиста "Ремонт PC своими руками" в журнале "РАДИО" номер 10 за 1991 год.
СТОИМОСТЬ — 3800 руб. + 28%

**Продаем программу увеличивающую емкость винчестера в два раза !**
Аппаратных доработок не требуется! ВСЕГО — 2000 руб. + 28 %!

### А также информация по телекоммуникациям:

**Справочно-учебное пособие по работе с HAYES-совместимым модемом**
Описание стандартных АТ-команд модема.
Стоимость — 1000 руб. + 28%.

**Набор телекоммуникационных программ для модемов и факс-модемных плат**
Программы поддерживают различные протоколы и системы передачи данных. Часть программ русифицирована.
Стоимость набора — 5000 руб.+28%.

**Набор программ для организации узла электронной почты (BBS).** Стоимость набора — 5000 + 28 %.

**Методическое пособие по пошаговой инсталляции ОС NOVELL 2.15.** Всего 230 стр. Стоимость — 4000 + 28%.

---

Для получения необходимой вам информации следует перевести указанную в рекламе сумму на наш р/с и выслать в наш адрес заявку произвольной формы. Возможно предварительное ознакомление и личное получение. Иногородним клиентам заказы высылаются почтой. при покупке за наличный расчет предоставляется скидка от 20 до 50 %.

*Наши банковские реквизиты: МП "РОСК" р/с 644849 в Коммерческом банке "Бизнес" МФО 201638, г.Москва*

*Наш адрес: 129010 Москва, а/я 837, МП "РОСК"*
*Телефоны: (095) 168-53-06, 475-89-84 с 14 до 17 часов*

## АРМ

**001 OK**
Автоматизация работы отдела кадров предприятия.
Круг 14800р.*(1)

**002 BLANK**
Выписка бухгалтерских документов (счета, платежные поручения, платежные требования).
Круг 10450р.(1)

**003 POST**
Автоматизация расчетов с поставщиками.
Круг 31900р.(1)

**004 АРМ бухгалтера "Mini-GROSS"**
Синтетический и аналитический учет хозяйственной деятельности. Расчет баланса в любой момент, платежные документы.
PL group 21120р.*(1)

**005 АРМ брокера "КОРОНА"**
Просмотр котировок в формате любой биржи. Выбор товаров по сложному запросу. Оформление договоров.
PL group 14080р.*(1)

**006 АСУ "Жилье"**
Автоматизация работ, связанных с учетом и распределением жилищного фонда.
ООО "ИНТЕС" - *

## ТЕКСТ-ПРОЦЕССОРЫ

**007 ETXT**
Пакет для обработки текстовой и графической информации. Редактор текстов ET.EXE, графический редактор BWI, конвертор графических файлов CONV составляет комплексную систему для получения текстово-графических документов.
С.Дорошев Центр "ЭМИС" 1780р.*

## ИНСТРУМЕНТЫ

**008 MENU FOR PROGRAMMERS**
Организация диалога типа МЕНЮ из прикладных программ (MS Fortran, MS C, Turbo C, Turbo C++, Turbo Pascal).
А.Владимиров 3800р.

**009 KITE INTERFACE MAKER v1.0**
Библиотека текстового интерфейса для Turbo C, Turbo C++.
А.Коршунов 2200р.*(2)

**010 SCREEN MAGIC**
Инструментальное средство для разработки экранных форм, меню. Интерфейс с разными языками программирования.
MIGHTYSOFT, USA 3300р.*

**011 K1816**
Программа для написания и отладки программ для микроконтроллеров серии К1816.
Дроздов И.А. 2200р.

## ПРИКЛАДНЫЕ ПРГРАММЫ

**012 ENGINE**
Система оптимального проектирования однофазных асинхронных двигателей малой мощности.
Чаварга В.П. - *

## ЗАЩИТА

**013 PROTECTION SYSTEM**
Защита .EXE-файлов от несанкционированного копирования. 3 типа защиты.
HardSoft 3400р.(1)

**014 PSW1**
Парольная защита .EXE и .COM программ от запуска.
А.Змановский 1595р.(1)

**015 KEYDISK**
Создание ключевой дискеты и привязка к ней исполняемых программ.
HardSoft 1600р.(1)

**016 HARDLOCK**
Индивидуальный и многопользовательские пароли на жесткий диск.
HardSoft 1400р.(1)

## УТИЛИТЫ

**017 PT**
Оригинальная программа печати текстов.
А.Строкинов 165р.(1)

**018 CV**
Программа перекодировки текстов, набранных в таблицах СМ, ВТА терминалов, КОТ-8.
А.Капланов 220р.(1)

**019 SWITCH**
Программа-драйвер клавиатуры, позволяющая перепланировать клавиатуру, переключиться на любой национальный алфавит, приписать любой цифровой или буквенной клавише произвольный символ.
Чаварга В.П. 330р.(1)

**020 POISK-18**
8 цветов на экране дисплея, стандартная клавиатура, дисководы, работающие в 3,5 раза быстрее, печатающий принтер и многое другое ПЭВМ "ПОИСК-1"
Д.Коваленко 165р.(1)

## ГРАФИКА

**021 GRAPHIC INTERFACE**
Экспорт, импорт и программная поддержка работы с изображениями.
HardSoft 5300р.*

**022 MEDAL**
Программа формирования оригинал-макетов круглых и треугольных печатей.
HARDWARE: HPLaserJet, EGA/VGA.
Толмачев А.Н. - *

## ИГРЫ

**023 STRIP CROSS PLATE**
Эротическая головоломка.
А.А.А. 150р.(1)
доп. дискета с данными 100р.(1)

## Уважаемые читатели!

Обращаем ваше внимание на то, что условия сотрудничества с "HS Listing" опубликованы в 5 и 6 номерах журнала за 1992 год.

"HS Listing" продолжает прием материалов в раздел "HARDWARE".

Прошло 6 месяцев с момента выхода "HS Listing", и мы предлагаем вашему вниманию рейтинг представленных в "HS Listing" программ по результатам продаж. В скобках указано количество реализованных копий.

### *HS BOARD*

| | |
|---|---|
| 1.Strip Cross Plate | (82) |
| 2.Key Disk | (31) |
| 3.Protection System | (27) |
| 4.PDD | (17) |
| 5.Menu for Programmers | (14) |
| 6.HS Driver | (8) |
| 7.HardLock | (5) |
| 8.OK | (2) |
| 9.CL | (1) |
| 10.ATP | (1) |

### *Новые расценки "HS Listing"*

Размещение рекламы: 650 р. для индивидуальных про-давцов; 1200 р. для организа-ций. Обозначения: '-' -"HS Listing" продуктом не располагает; '*' - наличие демо-версии; '()' - коли-чество дискет, необходимых для размещения продукта (стоимость каждой дискеты - 30 руб).

Название ____________

Текст ____________

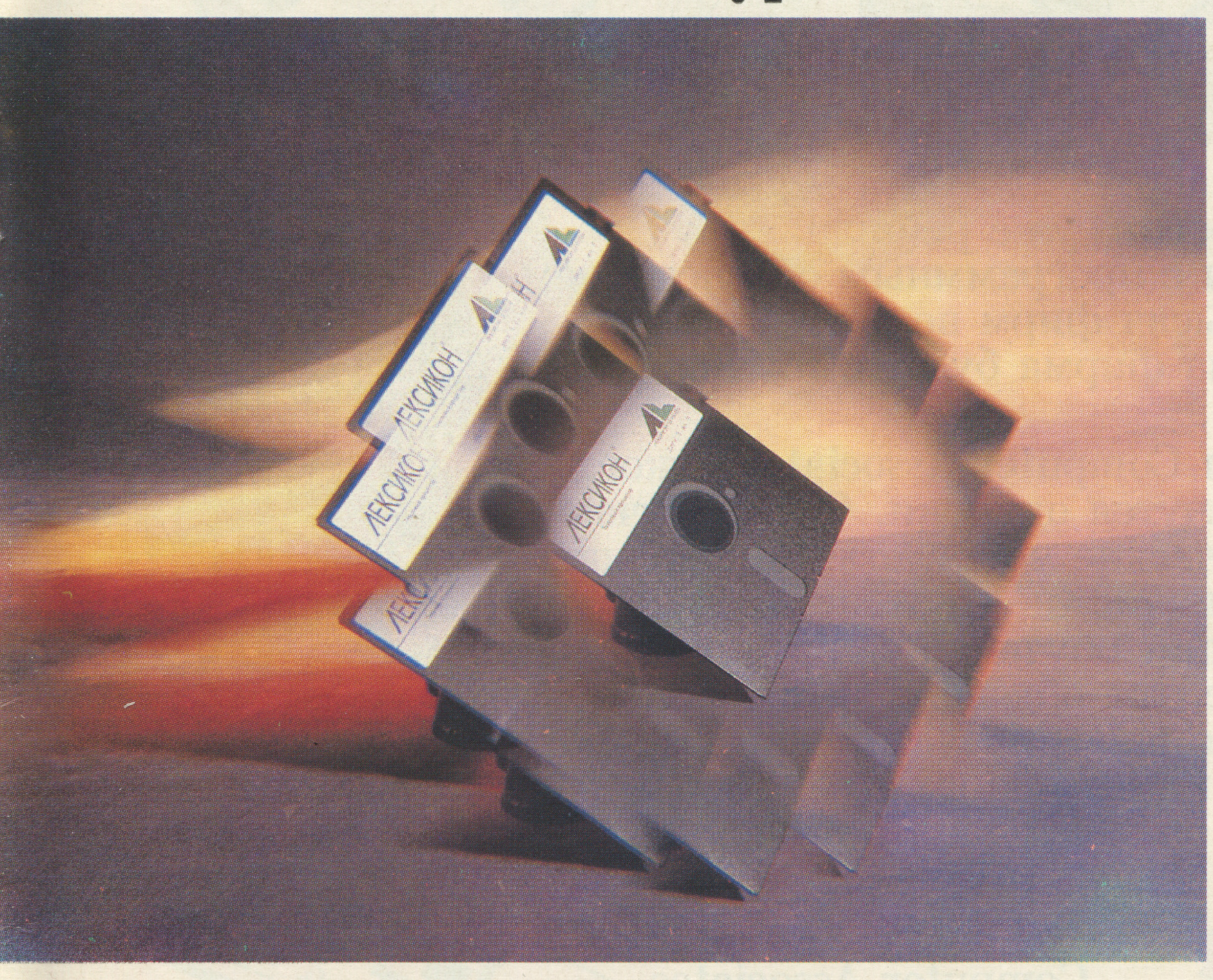
Автор ____________

Прошу распространять:
☑ Информацию
☐ Демо
☐ Программу Цена ________ р.

HS Listing
Адрес:109180,Москва,а/я34
Р/с 1609011 в Шаболовском
отд.МББ Москвы, код 2Е
МФО 201467 "Круг"

# ХОТИТЕ БЫТЬ УМНЫМ И БОГАТЫМ?

## Тогда для Вас — конкурс "А знаешь ли ты ЛЕКСИКОН?"




*В нем могут принять участие как юридичские, так и физические лица. Жюри конкурса будет оценивать точность и степень подробности ответов, присланных на конкурс. Первая премия — ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР!! (Только для зарегистрированных пользователей ЛЕКСИКОНа!) Десять поощрительных премий — пакет ЛЕКСИКОН последней версии с соответствующей регистрацией и дальнейшим сопровождением.*

**Вопросы конкурса:**

*1.* Дата начала распространения редактора ЛЕКСИКОН, автор разработки и номер первой версии.
*2.* Как напечатать с помощью ЛЕКСИКОНа один и тот же текст с различными межстрочными интервалами?
*3.* Как создать оглавление в начале документа, страницы которого должны быть автоматически пронумерованы?
*4.* Как нужно расставить страницы в тексте, чтобы не "разрезать" рисунки (псевдографику) на две страницы?
*5.* Что означает команда: >LEXICON -F4437.TMP -W3 TEST?
*6.* У Вас есть два принтера, подключенные к портам LPT1, LPT2. Как распечатать текст из ЛЕКСИКОНа по очереди на каждом из принтеров?
*7.* Как известно, любые часы хотя бы два раза в сутки показывают верное время. Сколько раз за время сеанса работы покажут верное время часы ЛЕКСИКОНа?
*8.* Найдите ошибки в ЛЕКСИКОНе 1.0 модификации 8.96 или 8.97.

Ответы присылайте до 1 февраля 1993 года по адресу: 113184 Москва, ул. Островского, 44. СП "Микроинформ". Телефон: (095) 233-00-06. Результаты конкурса будут опубликованы в нашем журнале в апреле 1993 года.

***Организаторы конкурса — СП "Микроинформ" и агентство "КомпьютерПресс"***









**Конкурс "А знаешь ли ты ЛЕКСИКОН?"**

Ф.И.О. ____________________

Организация ____________________

Адрес ____________________

____________________ тел.(___)__________

Ваши пожелания и предложения создателям ЛЕКСИКОНа ____________________

____________________

____________________

____________________



Крупнейший производитель
программного обеспечения в мире —
американская фирма Computer
Associates —
в России!
CA-Clipper Tools II — лучшая библиотека из 520 функций для Clipper 5.0.
Computer Associates уже решила
все проблемы с программным обеспечением
для сотен тысяч фирм во всем мире.
Пусть Ваша фирма будет в их числе.
Клиенты Computer Associates никогда
не останутся одиноки. Они получают
технические консультации ведущих
специалистов по бесперебойной телефонной
линии, обучение в сертифицированном
Центре обучения, возможность
демонстрировать свои разработки
на семинарах групп пользователей, участие
в международных конференциях.
CA-Clipper 5.0 — мировой стандарт систем управления базами данных.
Computer Associates —
это целая гамма
продуктов для DOS,
Windows,
и Macintosh
по самым низким ценам.
READ
@1,2GET Var1
JAN
FEB
DRAW
Computer Associates — это самый широкий набор
современных программных продуктов мирового класса
для больших, мини-ЭВМ и персональных компьютеров.
CA-SuperCalc — электронная таблица
будущего на Ваших
компьютерах
уже сегодня.
Информация о продуктах
Computer Assosiates,
демо-версиях, адрес
ближайшего к Вам дилера —
в представительстве Computer Assosiates Russia.
Россия, 127018, Москва, ул. 2-я Ямская, д.15.
Телефоны: (095) 289-44-77, 289-44-83.
Факс: (095) 289-34-69.